Resource Management - Data Access Design Document

EOEPCA.SDD.xxx

TVUK System Team

Version 0.1, dd/mm/yyyy:

Resource Management - Data Access Design Document

EO Exploitation Platform Common Architecture
Resource Management - Data Access Design Document
EOEPCA.SDD.xxx

COMMENTS and ISSUES	PDF
If you would like to raise comments or issues on this document, please do so by raising an Issue at the following URL https://github.com/EOEPCA/template-svce/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	dd/mm/yyyy	Initial in-progress draft

Chapter 1. Introduction

1.1. Purpose and Scope

This document presents the Resource Management - Data Access Design for the Common Architecture.

1.2. Structure of the Document

Section 2 - Overview

Provides an over of the Resource Management - Data Access component, within the context of the wider Common Architecture design.

Section 3 - Building Block Design

Provides the design of the Resource Management - Data Access component.

1.3. Reference Documents

The following is a list of Reference Documents with a direct bearing on the content of this document.

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

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

Reference	Document Details	Version
[OGC-WFS]	OGC Web Feature Service 2.0 Interface Standard – With Corrigendum, OGC 09-025r2, http://docs.opengeospatial.org/is/09-025r2/09-025r2.html	v2.0.2, 10/07/2014
[OGC-WCS]	OGC Web Coverage Service (WCS) 2.1 Interface Standard - Core, OGC 17-089r1, http://docs.opengeospatial.org/is/17-089r1/17-089r1.html	v2.1, 16/08/2018
[OGC-WCPS]	Web Coverage Processing Service (WCPS) Language Interface Standard, OGC 08-068r2, http://portal.opengeospatial.org/files/?artifact_id=32319	v1.0.0, 25/03/2009
[AWS-S3]	Amazon Simple Storage Service REST API, https://docs.aws.amazon.com/AmazonS3/latest/API	API Version 2006-03-01

1.4. Terminology

The following terms are used in the Master System Design.

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.

Term	Meaning
Application Package	A platform independent and self-contained representation of a software item, providing executable, metadata and dependencies such that it can be deployed to and executed within an Exploitation Platform. Comprises the *Application Descriptor* and the *Application Artefact*.
Bulk Processing	Execution of a *Processing Service* on large amounts of data specified by AOI and TOI.
Code	The codification of an algorithm performed with a given programming language - compiled to Software or directly executed (interpretted) within the platform.
Compute Platform	The Platform on which execution occurs (this may differ from the Host or Home platform where federated processing is happening)
Consumer	User accessing existing services/products within the EP. Consumers may be scientific/research or commercial, and may or may not be experts of the domain
Data Access Library	An abstraction of the interface to the data layer of the resource tier. The library provides bindings for common languages (including python, Javascript) and presents a common object model to the code.
Development	The act of building new products/services/applications to be exposed within the platform and made available for users to conduct exploitation activities. Development may be performed inside or outside of the platform. If performed outside, an integration activity will be required to accommodate the developed service so that it is exposed within the platform.
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

Term	Meaning
Home Platform	The Platform on which a User is based or from which an action was initiated by a User
Host Platform	The Platform through which a Resource has been published
Identity Provider (IdP)	The source for validating user identity in a federated identity system, (user authentication as a service).
Interactive Application	A stand-alone application provided within the exploitation platform for on-line hosted processing. Provides an interactive interface through which the user is able to conduct their analysis of the data, producing *Analysis Results* as output. Interactive Applications include at least the following types: console application, web application (rich browser interface), remote desktop to a hosted VM.
Interactive Console Application	A simple *Interactive Application* for analysis in which a console interface to a platform-hosted terminal is provided to the user. The console interface can be provided through the user's browser session or through a remote SSH connection.
Interactive Remote Desktop	An Interactive Application for analysis provided as a remote desktop session to an OS-session (or directly to a 'native' application) on the exploitation platform. The user will have access to a number of applications within the hosted OS. The remote desktop session is provided through the user's web browser.
Interactive Web Application	An Interactive Application for analysis provided as a rich user interface through the user's web browser.
Key-Value Pair	A key-value pair (KVP) is an abstract data type that includes a group of key identifiers and a set of associated values. Key-value pairs are frequently used in lookup tables, hash tables and configuration files.
Kubernetes (K8s)	Container orchestration system for automating application deployment, scaling and management.
Login Service	An encapsulation of Authenticated Login provision within the Exploitation Platform context. The Login Service is an OpenID Connect Provider that is used purely for authentication. It acts as a Relying Party in flows with external IdPs to obtain access to the user's identity.
EO Network of Resources	The coordinated collection of European EO resources (platforms, data sources, etc.).
Object Store	A computer data storage architecture that manages data as objects. Each object typically includes the data itself, a variable amount of metadata, and a globally unique identifier.
On-demand Processing Service	A *Processing Service* whose execution is initiated directly by the user on an ad-hoc basis.
Platform (EP)	An on-line collection of products, services and tools for exploitation of EO data

Term	Meaning
Platform Tier	The Platform Tier represents the Exploitation Platform and the services it offers to end-users
Processing	A set of pre-defined activities that interact to achieve a result. For the exploitation platform, comprises on-line processing to derive data products from input data, conducted by a hosted processing service execution.
Processing Result	The *Products* produced as output of a *Processing Service* execution.
Processing Service	A non-interactive data processing that has a well-defined set of input data types, input parameterisation, producing *Processing Results* with a well-defined output data type.
Products	EO data (commercial and non-commercial) and Value-added products and made available through the EP. *It is assumed that the Hosting Environment for the EP makes available an existing supply of EO Data*
Resource	A entity, such as a Product, Processing Service or Interactive Application, which is of interest to a user, is indexed in a catalogue and can be returned as a single meaningful search result
Resource Tier	The Resource Tier represents the hosting infrastructure and provides the EO data, storage and compute upon which the exploitation platform is deployed
Reusable Research Object	An encapsulation of some research/analysis that describes all aspects required to reproduce the analysis, including data used, processing performed etc.
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.

Term	Meaning
User	An individual using the EP, of any type (Admin/Consumer/Expert/Guest)
Value-added products	Products generated from processing services of the EP (or external processing) and made available through the EP. This includes products uploaded to the EP by users and published for collaborative consumption
Visualisation	To obtain a visual representation of any data/products held within the platform - presented to the user within their web browser session.
Web Coverage Service (WCS)	OGC standard that provides an open specification for sharing raster datasets on the web.
Web Coverage Processing Service (WCPS)	OGC standard that defines a protocol-independent language for the extraction, processing, and analysis of multi-dimentional coverages representing sensor, image, or statistics data.
Web Feature Service (WFS)	OGC standard that makes geographic feature data (vector geospatial datasets) available on the web.
Web Map Service (WMS)	OGC standard that provides a simple HTTP interface for requesting geo-registered map images from one or more distributed geospatial databases.
Web Map Tile Service (WMTS)	OGC standard that provides a simple HTTP interface for requesting map tiles of spatially referenced data using the images with predefined content, extent, and resolution.
Web Processing Services (WPS)	OGC standard that defines how a client can request the execution of a process, and how the output from the process is handled.
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

Term	Definition
DAL	Data Access Library
EMS	Execution Management Service
EO	Earth Observation
EP	Exploitation Platform
FUSE	Filesystem in Userspace
GeoXACML	Geo-specific extension to the XACML Policy Language
IAM	Identity and Access Management
IdP	Identity Provider
JSON	JavaScript Object Notation
K8s	Kubernetes
KVP	Key-value Pair
M2M	Machine-to-machine
OGC	Open Geospatial Consortium
PDE	Processor Development Environment
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PIP	Policy Information Point
RBAC	Role Based Access Control
REST	Representational State Transfer
SSH	Secure Shell
TOI	Time of Interest
UMA	User-Managed Access
VNC	Virtual Network Computing
WCS	Web Coverage Service
WCPS	Web Coverage Processing Service
WFS	Web Feature Service
WMS	Web Map Service
WMTS	Web Map Tile Service
WPS	Web Processing Service
WPS-T	Transactional Web Processing Service
XACML	eXtensible Access Control Markup Language

Chapter 2. Overview

TBD

Chapter 3. Building Block Design
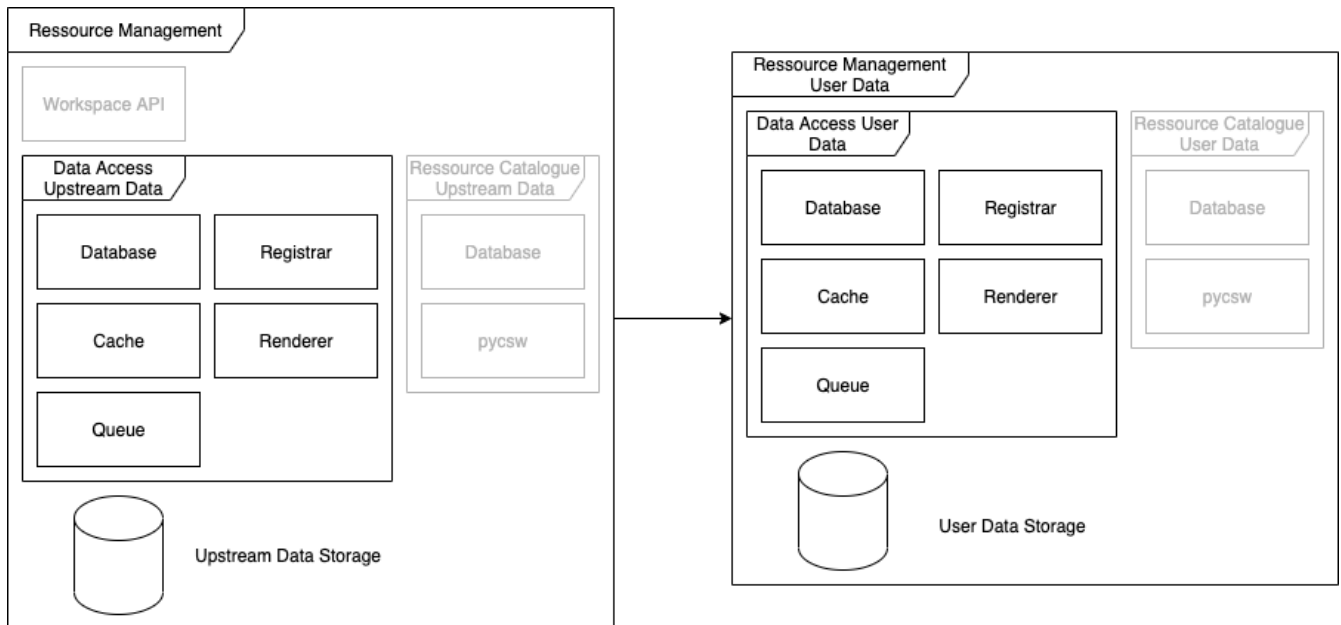
When discussing the Data Access Building Block it needs to be stressed, that there are actually two kinds of deployments of the same software stack in two slightly different roles.

The first role is sort of system global instances, registering and serving upstream data products whereas the second one is used within the context of a user workspace. In this second context, only user provided data will be registered and served.

The global instance is installed in a global namespace using a GitOps approach, whereas the user workspace deployment of the Data Access Service is deployed via the Workspace API dynamically.

3.1. Registrar Service

3.1.1. Overview and Purpose

The purpose of the registrar service is to register data products from either upstream data sources (such as Sentinel-2 data archives) into the Data Access Building Block and the Resource Catalogue Building Block where they are indexed for later access.

3.1.2. Software Reuse and Dependencies

This service is based upon the Registrar Service of the View Server software system, using its plugin system to allow for a registration also into the Resource Catalogue.
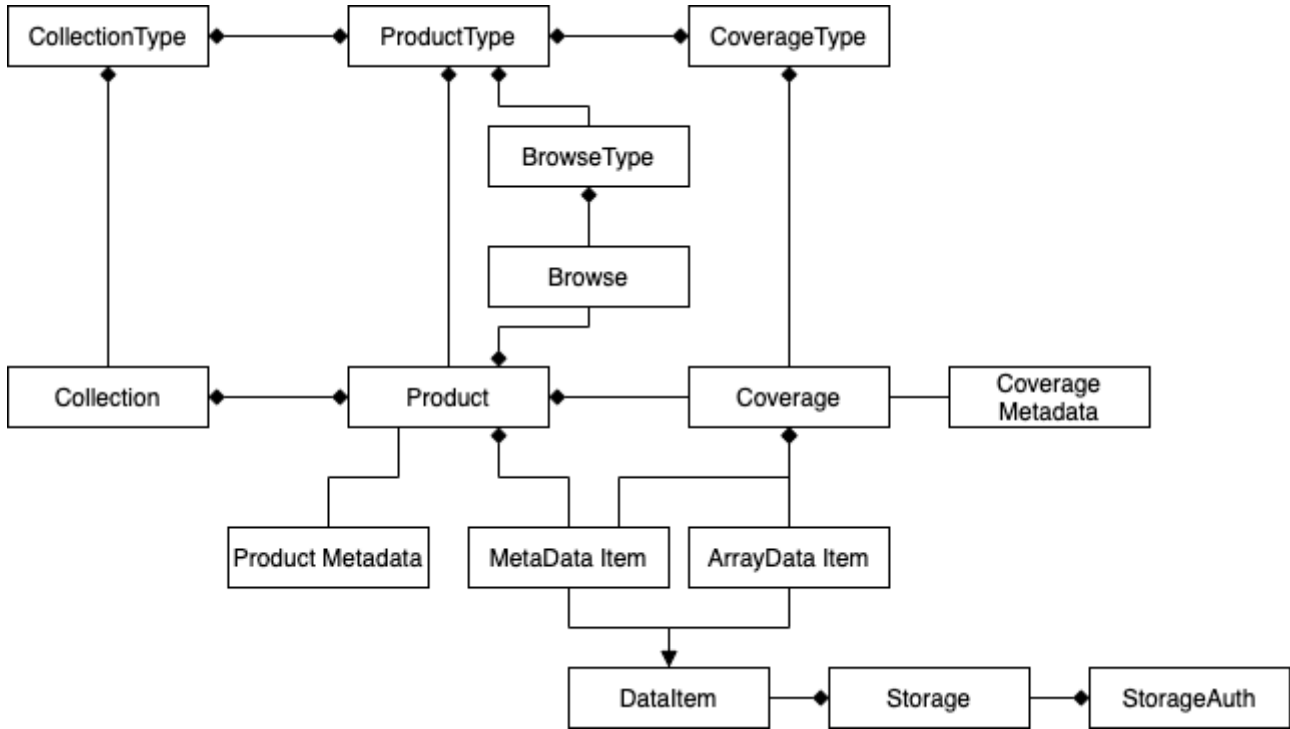
3.1.3. Interfaces

This service does not provide any interfaces.

3.1.4. Data

3.1.4.1. Data Model

The following image shows the registrars (and also the renderers) internal data model:

3.1.4.1.1. Collection/CollectionType

Collections are used to form variant groupings of products and/or coverages. They can either be homogenous, semi-homogenous or non-homogenous, depending on the configuration of the Collections associated CollectionType. The CollectionType can limit the insertion of Products and Coverages by specifying allowed Product-/CoverageTypes, which are then the only types of Products/Coverages that can be inserted.

The spatio-temporal metadata (time-span and footprint) is always the summary of all entailed items.

3.1.4.1.2. Product/ProductType

A Product is a collection of spatio-temporally and acquisition related data items. Each Product has a unique identifier, a spetio-temporal coverage and optionally additional metadata associated.
Each Product is of a specific ProductType, limiting in which Collections it can be inserted and what Coverages can be part of a Product.

3.1.4.1.3. Coverage/CoverageType

A Coverage is an n-dimensional construct of a specific grid. It consists of at one data file that stores the actual values of the Coverage. Each Coverage is associated with a CoverageType, adding additional metadata to the internal structure of each such Coverage and limiting the possible Products this Coverage can be assoctiated with. Each Coverage can only be part of one Product.

3.1.4.1.4. Browse/BrowseType

A Browse is a visual representation of a Product and the BrowseType is the blueprint to create such a representation of a given Product. Each Browse is associated with a BrowseType but it is not required for each Product to have a Browse of each available BrowseType. In this case, the browse is considered virtual and the representation will be generated from the Products associated Coverages every time.

The BrowseType consists of expressions and ranges that allow to generate the greyscale, RGB or RGBA images.

3.1.4.1.5. Product/Coverage Metadata

This model store additional Product/Coverage level metadata (such as snow- or cloud coverage) that can be used in filters.

3.1.4.1.6. ArrayData Item

This is a link to a "file" containing the raster data for a specific Coverage.

3.1.4.1.7. MetaData Item

This is a link to a file containing Product/Coverage metadata.

Each DataItem can be associated with a Storage, which specifies the actual location of the DataItem. If no Storage is specified the DataItem points to a file on the local filesystem.

3.1.4.1.8. Storage

A Storage describes a physical storage location such as an HTTP server, FTP service or Object Storage of various types. For services requiring more elaborate authentication/authorization the Storage must be linked to a StorageAuth of a compatible type.

3.1.4.1.9. StorageAuth

This model stores the authentication/authorization values for any number of Storages. This may include, but is not limited to access tokens, access credentials, public/private keys, etc.

3.1.4.2. Configuration

This service uses a YAML based configuration to allow the highly configurable configuration from various data sources parsed using various schemas into the backends.

3.1.4.3. Data flow

The registrar is a server process, listening on a distributed queue for new registration requests coming in as paths to the respective item to be registered.
When such a registration request is passed via the queue, the respective data storage source will be determined and the relevant data and metadata files are read for a rudimentary classification of the to be registered item, which will be passed to all configured backends for a final registration of the item.

The registrar allows for various item schemes:

- Sentinel-2 L1C/L2A: This expects an unpacked Sentinel-2 Level 1C or 2A SAFE package as the path input. It will parse all product and granule related metadata and will register the product as a single item. Each Product will be registered for the Data Access building block in the following manner:

  - Product: the whole SAFE package will be referenced as a single Product. The root and tile metadata files are associated as its MetaData items. The ProductType is the pre-created type for either Sentinel-2 L1C or L2A.

  - Coverage: each raster data file in the scene will be added as a single Coverage, using the Products identifier as a common base name and using the band identifier as a postfix. This includes the following bands: B01, B02, B03, B04, B05, B06, B07, B08, B8A, B09, B10 (only L1C), B11, B12. In case of L2A, only the band with the highest resolution will be used. The CoverageType for each Coverage is the pre-registered band specific CoverageType for the used ProductType.

  - Browse: The TCI image of each SAFE will be registered as the Products default representation. For L2A, only the highest resolution TCI is used. Some pre-registered BrowseTypes (TRUE_COLOR, FALSE_COLOR, and NDVI) are used in a virtual manner.

- STAC-Catalog: This scheme requires a path to a directory containing at least one `catalog.json` file that will be read and parsed for both sub-catalogs and/or referenced STAC items. All STAC items found this way will be registered into all the backends.

  - The Items Product-/CoverageTypes will be looked up using the STAC Items asset identifiers or created if necessary.

  - Product: each STAC Item is translated to a Product, using the Items identifier as the Product identifier. The product type will be the one earlier looked up or created.

  - Coverage: each associated asset in the STAC Item having a `eo:band` property will be added as a Coverage, using the Products identifier as a prefix and the asset identifier as a postfix.

3.1.5. Applicable Resources

https://docs.eoxserver.org/en/master/

3.2. Renderer Service

3.2.1. Overview and Purpose

This service allows to generate automatic renderings of the registered data items via various standardized interfaces.

3.2.2. Software Reuse and Dependencies

This service is based upon the Renderer Service of the View Server software system with only minor enhancements.

3.2.3. Interfaces

3.2.3.1. OGC Web Service (OWS) interfaces

The renderer service provides various OGC compliant service endpoints, to enable the creation of dynamic renderings/processings of the referenced datasets.

3.2.3.1.1. Web Coverage Service (WCS)

With the OGC Web Coverage Service (WCS) interface, users can access the raw raster data values of the stored Earth Observation data or coverages. A coverage is a multidimensional spatio-temporal object and can be subset along any axis and/or field.
With the EO extension (EO-WCS), it is possible to define homogeneous or heterogeneous collections (Stitched Mosaics and Dataset Series respectively) of EO metadata enriched coverages that can be searched in time and space.

In this service, each Collection is represented as a `DatasetSeries`, which can be queried using the `DescribeEOCoverageSet`. Each Collection is thus advertised in the Capabilities document of the service.

Each Product is also represented as a `DatasetSeries` but in contrast to Collections not advertised in the Capabilities document.

Each Coverage is represented as a Coverage directly, but not listed in the Capabilities document.

3.2.3.1.2. Web Map Service (WMS)

The OGC Web Map Service (WMS) interface standard provides rendered maps (images) to be displayed in the users' graphical user interface or similar application. This interface revolves around the concept of the layer, from which subsets can be fetched. This can be static data prepared for each request or can have dynamic filters such as specific rendering instructions or data filters applied.
The Earth Observation Application Profile (EO-WMS) provides guidance how to apply WMS on Earth Observation data. For example, it details how to provide a collection or dataset like the whole Sentinel-2 archive as one WMS layer. Subsets down to individual products can be visualized using the TIME dimension or parameter.
Additional extension exists via custom or vendor specific parameters. One such extension is the CQL (Common Query Language) parameter as used in ESA's PRISM activity for example to select individual products based on ID or to filter on additional parameters like cloud coverage.

Each registered Collection is represented as a hierarchy of layers in the following form:

- A root layer with the same name as the Collection with the following sub-layers with the associated suffixes:

  - The Root layer is rendered using the default Browse representation of each Product.

  - Outlines (`···_outlines`): This shows the outlines of each Product in the collection.

  - Outlined root layer (`···_outlined`): This shows the outlines of each Product in the collection overlayed over the default browse representation.

○ Browse Type (···_`<browse-type-name>`): this renders each product in the collection with the specified browse type. When a pre-rendered browse is available it is used, otherwise the dynamic rendering process is used.

Each Product is also represented in the same structure as Collections with the sole difference that they are not by default represented in the Capabilities document, only when a CQL query using the `cql` parameter matches this Product is passed.

3.2.3.1.3. OpenSearch

OpenSearch is a collection of simple formats for the sharing of search results. The OpenSearch description document format can be used to describe a search engine so that it can be used by search client applications. The OpenSearch description format allows the use of extensions that allow search engines to request a specific and contextual query parameter from search clients. The OpenSearch response elements can be used to extend existing syndication formats, such as RSS and Atom, with the extra metadata needed to return search results.

The CEOS OpenSearch Best Practice Document is providing server implementation best practices for EO OpenSearch search services that allow for standardized and harmonized access to metadata and data for CEOS agencies. Within this context, the following OGC extensions and recommendations are applicable:

- OpenSearch GEO: OpenSearch Geo and Time Extensions

- OpenSearch EO: OGC OpenSearch Extension for Earth Observation

The OGC OpenSearch Geo and Time standard specifies the Geo and Time extensions to the OpenSearch query protocol to geographically constrain search results.

The Earth Observation Extension specifies a series of parameters that can be used to constrain search results. In short, provision is made to filter results by sensor information, acquisition, processing parameters and other information. The purpose of the OpenSearch Extension for Earth Observation is to make sure that OpenSearch parameters are aligned with OGC 10-157r4 that describes EO products metadata and with ISO19115(-1)/ISO19115-2 that is used for describing EO collection metadata. In a typical search scenario, a client will first search for the appropriate EO Collection with the parameters appropriate to EO Collections. In the search response he will find the details (e.g. the identifier or the link to the OpenSearch description document) to search for EO Products of that EO Collection that he identifies as most appropriate.

OGC defines a GeoJSON and JSON-LD encoding standard of Earth Observation (EO) metadata for collections (a.k.a. dataset series). The standard provides document models for the exchange of information describing EO collections, both within and between different organisations.

EO collections are collections of datasets sharing the same product specification. These collections are also called dataset series as they may be mapped to 'dataset series' following the terminology defined in ISO 19113, ISO 19114 and ISO 19115. An EO collection typically corresponds to a series of EO datasets (also called EO products) derived from data acquired:

- Either from an instrument in a dedicated mode on board a single satellite platform; or

- by a series of instruments, possibly from different satellite platforms, but in this case working in the same instrument mode

In this service, Collections are searched in the first step of the two-step search, whereas Products

within those Collections are searched in the second step.

3.2.3.1.4. Admin Interface

The admin interface allows operators to inspect and ultimately alter the internal data models of the service. The interface is structured very similarly to the model layout as detailed in the registrar section.

3.2.4. Data

3.2.4.1. Configuration

The application configuration is stored in the Database service where, depending on the request, all relevant metadata is extracted and used in the rendering process.

3.2.4.2. Data flow

As a web service, the renderer awaits user requests which are then processed. For that, initial queries to the database service are made, which in turn deliver the information of what files are required to fulfill the request. In the ensuing process, these files, residing on an object storage or a mounted network file system are then accessed and the required portions extracted. Finally, the resulting image, or data files are returned to the caller.

3.2.5. Applicable Resources

https://docs.eoxserver.org/en/master/

3.3. Cache Service

3.3.1. Overview and Purpose

The purpose of this service is to provide a caching layer for WMS interface of the renderer service, as they may be computationally costly to produce.
Caching can happen either beforehand (pre-seeded) or on demand (or a mixture of both), in order to even further improve performance, even for the first lookup.
Caching is performed on a tile basis for each registered dataset, using the time axis to distinguish the individiual scenes in a collection. In order to resolve the time axis, a connection to the database service is used.

3.3.2. Software Reuse and Dependencies

This service is realized using the COTS MapCache with a custom confugation.

3.3.3. Interfaces

This service exposes the WMS and WMTS OGC Web Services endpoints.

3.3.3.1. OGC Web Service (OWS) interfaces

The Cache Service provides various OGC Web Service endpoints. The provided layers use a static configuration, mimicking the dynamic status of the contents of the Renderer Service.

3.3.3.1.1. Web Map Tile Service (WMTS)

The OGC Web Map Tile Service (WMTS) interface standard is very similar in nature to the WMS, as it provides rendered images of stored data. In contrast, however, only tiles of one of the pre-defined tile grids can be accessed and only in a limited number of predefined styles.
This way, the tiles can be efficiently pre-processed and cached, allowing for better performance when accessing the service.
The EO-WMS guidance can be applied on WMTS as well for example to support the TIME dimension on collection level layers.

3.3.3.1.2. Web Map Service (WMS)

The OGC Web Map Service (WMS) interface standard provides rendered maps (images) to be displayed in the users' graphical user interface or similar application. This interface revolves around the concept of the layer, from which subsets can be fetched. This can be static data prepared for each request or can have dynamic filters such as specific rendering instructions or data filters applied.
The Earth Observation Application Profile (EO-WMS) provides guidance how to apply WMS on Earth Observation data. For example, it details how to provide a collection or dataset like the whole Sentinel-2 archive as one WMS layer. Subsets down to individual products can be visualized using the TIME dimension or parameter.

3.3.4. Data

3.3.4.1. Configuration

A single configuration file defines the cache behavior. The structure of this XML based configuration file can be inspected on the MapCache homepage.

3.3.4.2. Data flow
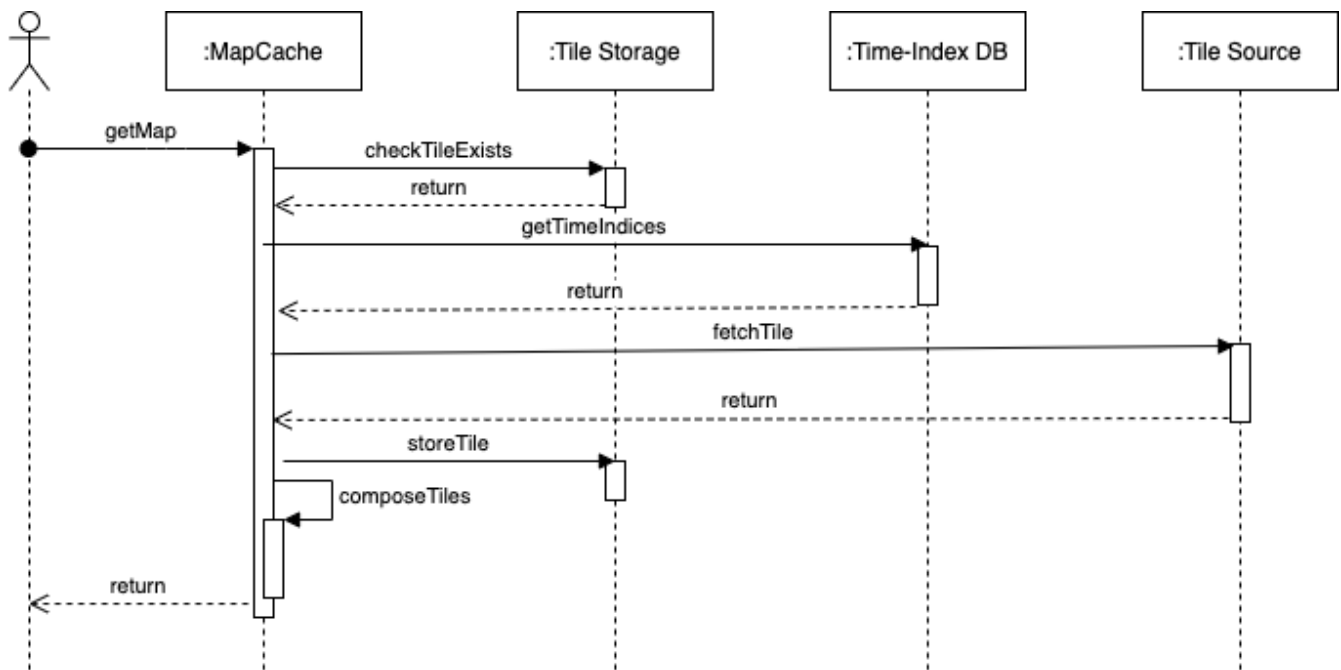
Similarly to the renderer service, the cache service exposes an HTTP endpoint that dispatches requests for the provided OGC Web Services. Depending on the request, a database query may be involved in order to resolve the time axis.

Now it is checked, whether the tiles involved with the request are already cached or need to be rendered by the renderer service. Each tile that is missing in the cache is now requested from the renderer and subsequently cached in the backend, the configured object storage.

The final response is now merged from all intersecting tiles and returned to the client.

3.3.5. Applicable Resources

The MapCache documentation can be found here: https://mapserver.org/mapcache/

3.4. Client Service

3.4.1. Overview and Purpose

This service provides a configured client to be run in a browser.

3.4.2. Software Reuse and Dependencies

The server software used is the open source software nginx, serving a pre-built and configured JavaScript application eoxc, which is in turn based on the mapping library OpenLayers.

3.4.3. Interfaces

This service provides an HTTP endpoint to retrieve the client files.

3.4.4. Data

3.4.4.1. Configuration

The client is configured using JavaScript inside the main index.html.

3.4.4.2. Data flow

When requested, the client JavaScript bundle is downloaded by the browser and the application is initialized. This application will connect to the endpoints of various services such as the cache and renderer, but also external sources for map base-, or overlay layer tiles. The requested map tiles and metadata will be visualized within the app or made available as a downloaded file.

3.4.5. Applicable Resources

The git repository with additional resources for EOxC can be found at the projects repository: https://github.com/eoxc/eoxc

3.5. Database Service

3.5.1. Overview and Purpose

This service provides the main database facilities for the other services requiring relational table storage.

3.5.2. Software Reuse and Dependencies

This service is using the COTS version of the popular PostgreSQL system software.

3.5.3. Interfaces

This service provides a single TCP based access mechanism through which commands can be executed.

3.5.4. Data

3.5.4.1. Configuration

This service is configures via a configuration file.

3.5.4.2. Data flow

3.5.5. Applicable Resources

The documentation for PostgreSQL can be found here: https://www.postgresql.org/docs/

3.6. Queue Service

3.6.1. Overview and Purpose

This service serves as a central point of communication between the services of the data access building block. Various sets and lists are used to track incoming registration requests and their subsequent status.

3.6.2. Software Reuse and Dependencies

This service is a configured instance of the Redis COTS software.

3.6.3. Interfaces

This service provides a TCP based endpoint for all commands.

3.6.4. Data

3.6.4.1. Configuration

No additonal configuration is used beyond the default settings.

3.6.4.2. Data flow

`registration_queue`: this list based queue is used to buffer incoming registration requests. It is used as a FIFO (first-in-first-out) queue, so the earlier registration request is handled first.

`registered_set`: This set of strings collects all registration items that were successfully registered.
`failure_set`: This set contains all the paths of items that failed to register.

3.6.5. Applicable Resources

https://redis.io/documentation

<< End of Document >>