

# Engineering Objects (Draft)

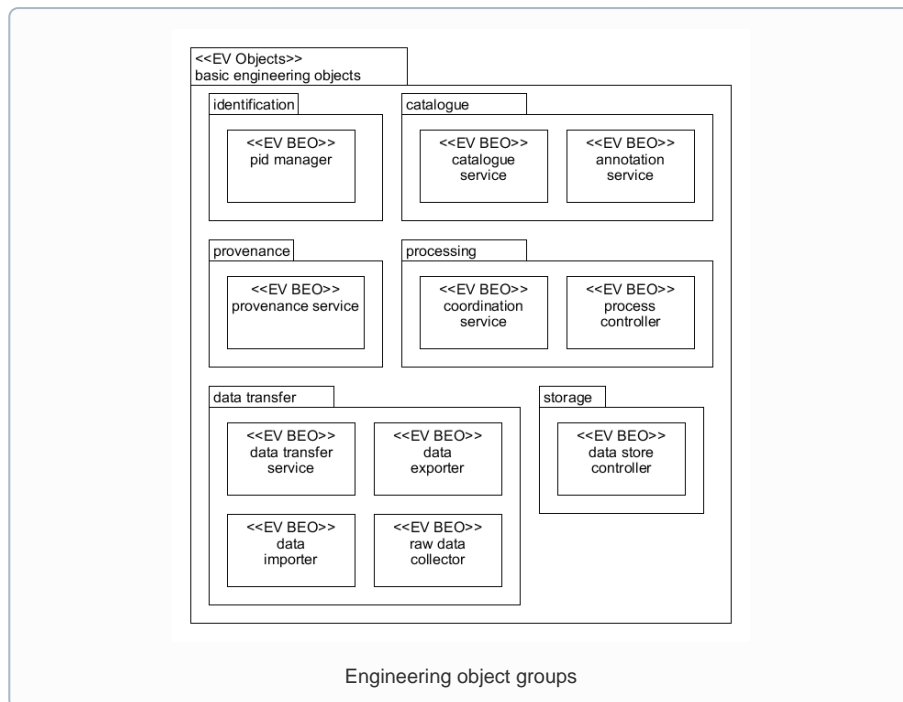
The objects described by the engineering viewpoint directly support research data management functionalities which are commonly shared among more than one RI (Identification and Citation, Curation, Cataloguing, Processing, and Provenance) <sup>01, 02</sup>.

The engineering objects are grouped in the four top packages shown in the figure to the right: identification, catalogue, processing, and provenance, in line with the support needed for the four common data management functionalities. Two additional groups are defined to accommodate supporting objects which are needed for any working configuration: storage and data transfer. The objects included are defined as **Basic Engineering Object (BEO)** and provide functionalities described by **CV Objects**, this means that each **BEO** maps one to one to a **CV Object**.

The grouping is not strict, these objects can be deployed side by side on a single physical machine or be distributed across network and institutional limits.

The engineering objects coupled with the **Container Structure**, are used to further define how the engineering objects can support the data management functionalities in different **Object Configuration**.

The definition of the basic engineering objects are given as follows. All of the definitions with the exception of pid manager and provenance service, are derived from the definitions of the corresponding computational viewpoint objects.



## annotation service

A basic engineering object that supports the ability associate annotations (free texts and semantic) with the assets managed by a research infrastructure. An annotation service must provide three functionalities: annotate data, update catalogues and update records

Annotation service functions

Function	Description	Type
annotate data	Function for requesting the annotation (free text or semantic) of existing assets. This may require the creation of additional records to record qualitative observations, provenance or structured metadata.	server
update catalogues	Function for updating of annotation metadata and the associations to existing assets.	client
update records	Function for updating of annotation records associated to assets.	client

Annotation service correspondences

Viewpoint	Correspondence
Science	<a href="#">data annotation</a> (behaviour)
Information	<a href="#">metadata catalogue</a> (instance of object)
Computational	<a href="#">annotation service</a>

## catalogue service

A basic engineering object that supports the ability to publish and search collections of metadata (descriptive information) for assets such as data, services, and related information objects managed by a research infrastructure. A catalogue service must provide at least four functionalities: query catalogues, update catalogues, export metadata, and query resource.

Catalogue service functions

Function	Description	Type
----------	-------------	------

Catalogue service correspondences

Viewpoint	Corresponding Object
-----------	----------------------

export metadata	function for gathering metadata to be exported with assets extracted from the data curation store objects (data stores).	server
query catalogues	function for querying assets held by the infrastructure, including the retrieval of assets associated with a given persistent identifier.	server
update catalogues	function for updating of catalogues and the associated data assets.	server
invoke resource	function for enabling the invocation of other services such as harvesting, exporting data, or automated update. This includes the communication with internal components such as the data store controller for retrieving data.	client

## coordination service

A basic engineering object that that supports the ability to coordinate data processing tasks on infrastructure execution resources. A coordination service should provide at least three functions: process request, coordinate process, create process controller, and prepare data transfer

Coordination service functions

Function	Description	Type
process request	function for scheduling the execution of data processing tasks. This could require executing complex workflows involving many (parallel) sub-tasks.	server
coordinate process	function used to coordinate the execution of data processing tasks on execution resources presented by process controllers.	server
create process controller	function for creating and configuring the required <a href="#">process controllers</a> required for the different processing tasks	internal
prepare data transfer	function used to retrieve data (and metadata) from the data stores and to return the processing results.	server

## data exporter

A basic engineering object for exporting assets from the data stores. A data exporter should provide five functions: retrieve asset, update records, export metadata, export asset, deliver asset.

data exporter functions

Function	Description	Type
retrieve asset	function provided to enable requesting data from a designated data store (inherited from <a href="#">data transporter</a> ).	client
update records	function provided to inform downstream resources about impending asset transfers (inherited from <a href="#">data transporter</a> ).	client
export metadata	function provided for requesting any additional metadata associated with the asset being transferred.	client
export asset	function provided for extracting assets from a designated data store	consumer
deliver asset	function provided for delivery of assets to a designated data store.	producer

## data importer

A basic engineering object for importing assets into the data stores. A data importer should provide six functions: retrieve asset, update records, export metadata, export asset, deliver asset.

data importer functions

Function	Description	Type
----------	-------------	------

Science	<a href="#">catalogue system</a> (role)
Information	<a href="#">metadata catalogue</a> (object)
Computational	<a href="#">catalogue service</a>

Catalogue service correspondences

Viewpoint	Corresponding Object
Science	<a href="#">data processing subsystem</a> (role) <a href="#">coordinate service</a> (behaviour)
Information	<a href="#">process data</a> (action)
Computational	<a href="#">coordination service</a>

data exporter correspondences

Viewpoint	Corresponding Object
Science	<a href="#">data publishing subsystem</a> (role) <a href="#">data use subsystem</a> (role)
Computational	<a href="#">data exporter</a>

data importer correspondences

Viewpoint	Corresponding Object
-----------	----------------------

retrieve asset	function provided to enable requesting data from a designated data store (inherited from <a href="#">data transporter</a> ).	client
update records	function provided to inform downstream resources about impending asset transfers (inherited from <a href="#">data transporter</a> ).	client
acquire identifier	function provided for requesting new persistent identifier to be associated with the data being transferred. Generally, identifiers are requested when importing new data into an infrastructure.	client
update catalogues	function for populating the catalogues with the metadata associated to the imported data assets.	client
import asset	function provided for receiving assets from a designated data store	consumer
deliver asset	function provided for delivery of assets to a designated data store.	producer

Science	<a href="#">data acquisition subsystem</a> (role) <a href="#">data curation subsystem</a> (role)
Computational	<a href="#">data importer</a>

## data store controller

A basic engineering object that encapsulates the functions required to store and maintain data assets managed by the RI. The data store controller also provides access to authorised agents. A data store controller should provide five functions: update records, query resource, retrieve data, import data, export data.

data store controller functions

Function	Description	Type
update records	function for editing data records within a data store, as well as preparing ingestion of data streams.	server
query resource	function for querying the data store.	server
retrieve data	functions to manage the retrieval of assets from a data store	server
import asset	function provided for receiving assets into a designated data store	consumer
export asset	function provided for extracting assets from a designated data store.	producer

data store controller correspondences

Viewpoint	Corresponding Object
Science	<a href="#">data curation subsystem</a> (role)
Computational	<a href="#">data store controller</a>

## data transfer service

A basic engineering object for managing the transfer of assets into and out-of the data stores. The data transfer service is responsible for setting up data transfers, including any repackaging of assets necessary prior to delivery.

data transfer service functions

Function	Description	Type
prepare data transfer	function for managing and scheduling a data transfer either into or out of the data stores	server
create transporter	function for creating and configuring the required data transporters ( <a href="#">raw data collector</a> , <a href="#">data importer</a> , or <a href="#">data exporter</a> )	internal

data transfer service correspondences

Viewpoint	Corresponding Object
Science	<a href="#">data curation subsystem</a> (role)
Computational	<a href="#">data transfer service</a>

## pid manager

A basic engineering object for managing the acquisition, registration, and maintenance of persistent identifiers for data assets. Persistent identifiers can be generated internally or externally. For assigning resolvable global unique identifiers, the pid manager commonly depends of an external [PID service](#). A pid manager should provide three functionalities: acquire identifier, manage identifier, and resolve identifier.

pid manager functions

Function	Description	Type
acquire identifier	function for providing new persistent identifiers for assets	server

data transfer service correspondences

Viewpoint	Corresponding Object
Science	<a href="#">PID Manager</a> (role)
Information	<a href="#">assign unique identifier</a> (action)

manage identifier	function for retrieving, updating and deleting identifiers	server
resolve identifier	function for providing the physical location of an asset to authorised requester	server

Computational	<b>PID manager</b> (object)
---------------	-----------------------------

## process controller

A basic engineering object providing the functions required for using an execution resource (generically, any computing platform that can host some process) as part of a predefined workflow. A process controller should provide five functions:

process controller functions

Function	Description	Type
coordinate process	function for controlling the execution resource associated with a given process	server
retrieve results	functions for retrieving results from an execution resource	server
update records	provides functions for modifying data on an execution resource, including preparing the resource for the ingestion of bulk data streams	server
stage data	used to acquire streams of data from data store objects as part of some process.	consumer
export asset	function provided for retrieving the assets produced by some process	producer

Catalogue service correspondences

Viewpoint	Corresponding Object
Science	<b>data processing subsystem</b> (role) <b>compose service</b> (behaviour)
Information	<b>process data</b> (action)
Computational	<b>process controller</b>

## provenance service

A basic engineering object that supports the ability to publish and search collections of provenance metadata for assets such as data, services, and related information objects managed by a research infrastructure. A provenance service is a specialisation of a **catalogue service**. A provenance service must provide at least four functionalities: query catalogues, update catalogues, export metadata, and query resource.

Provenance service functions

Function	Description	Type
export metadata	function for gathering metadata to be exported with assets extracted from the data store objects.	server
query catalogues	function for querying assets held by the infrastructure, including the retrieval of assets associated with a given persistent identifier.	server
update catalogues	function for updating of catalogues and the associated data assets.	server
invoke resource	function for enabling the invocation of other services such as harvesting, exporting data, or automated update. This includes the communication with internal components such as the data store controller for retrieving data.	client

Catalogue service correspondences

Viewpoint	Corresponding Object
Science	<b>catalogue system</b> (role)
Information	<b>track provenance</b> (action) <b>metadata catalogue</b> (instance of object)
Computational	<b>catalogue_service</b> (instance of object)

## raw data collector

A basic engineering object to acquire, package, and transfer raw data produced by acquisition objects. A raw data collector should provide six functions: retrieve asset, update records, export metadata, export asset, deliver asset.

raw data collector functions

Function	Description	Type
retrieve asset	function provided to enable requesting data from a designated data store (inherited from <b>data transporter</b> ).	client
update records	function provided to inform downstream resources about impending asset transfers (inherited from <b>data transporter</b> ).	client

raw data collector correspondences

Viewpoint	Corresponding Object
Science	<b>data acquisition subsystem</b> (role) <b>data curation subsystem</b> (role)
Computational	<b>raw data collector</b>

acquire identifier	function provided for requesting new persistent identifier to be associated with the data being transferred. Generally, identifiers are requested when importing new data into an infrastructure.	client
update catalogues	function for populating the catalogues with the metadata associated to the imported data assets.	client
import asset	function provided for retrieving raw data from data acquisition objects	consumer
deliver asset	function provided for delivery of assets to a designated data store.	producer

<sup>01</sup> ENVRIplus. (2016) Deliverable 5.1 A consistent characterisation of existing and planned RIs, ENVRI plus, 24/05/2016, Horizon 2020 Grant Agreement No 654182 <http://www.envriplus.eu/wp-content/uploads/2016/06/A-consistent-characterisation-of-RIs.pdf>

<sup>02</sup> ENVRIplus. (2017) Deliverable 8.3 Interoperable cataloging and harmonization for environmental RI projects: system design. ENVRI plus, 30/01/2017, Horizon 2020 Grant Agreement No 654182 <http://www.envriplus.eu/wp-content/uploads/2015/08/D8.3-Interoperable-cataloging-and-harmonization-for-environmental-RI-projects-system-design.pdf>