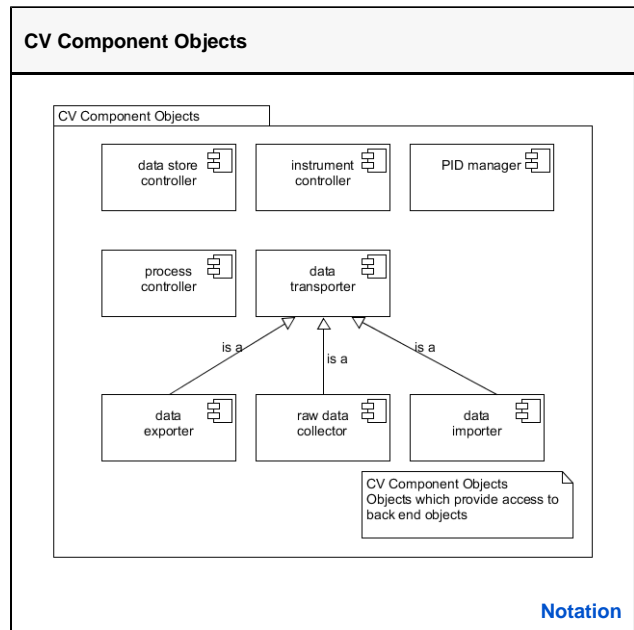


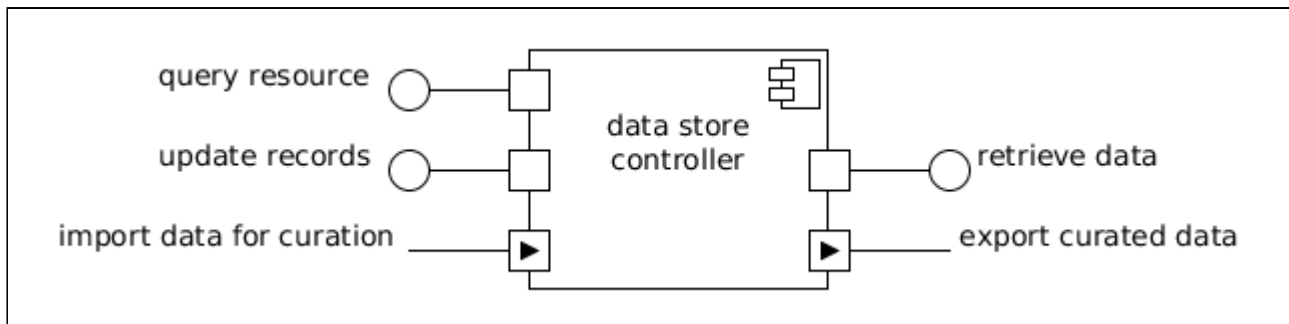
CV Component Objects

CV component objects offer programmatic access to the actual RI's systems and resources, the back end objects. This allows providing intermediate façades for systems and resources which may be interchanged or replaced as needed.

- **Data store controllers** provide access to data stores that may have their own internal data management regimes.
- **Instrument controllers** encapsulate the accessible functionalities of instruments and other raw data sources out in the field.
- **Process controllers** represent the computational functionality of registered execution resources.
- **data_transporter** are provided for managing the movement of data from one part of a research infrastructure to another.
 - **Raw data collectors** manage the movement of data from one or more data acquisition objects to one or more data store objects.
 - **Data importers** manage the movement of data from external sources (such as user-originated datasets and derived datasets from data processing) to one or more data stores objects.
 - **data_exporter** manage the movement of data from one or more data store objects to external destinations (such as a user machine or downstream service gathering data from the research infrastructure).
- **PID manager** handle the assignment, update, retrieval and deletion of persistent identifiers for data assets.



Data store controller



A data store supporting data preservation.

Data stores record data collected by the infrastructure, providing the infrastructure's primary resources to its community. A data store controller encapsulates the functions required to store and maintain datasets and other data artefacts produced within a data store of the RI, as well as to provide access to authorised agents.

A data store controller should provide three operational interfaces:

- **update records (server)** provides functions for editing data records within a data store as well as preparing a data store to ingest new data through its import stream interface described below.
- **query resource (server)** provides functions for querying the data held in a data store.
- **retrieve data (server)** provides functions to negotiate the export of datasets from a data store.

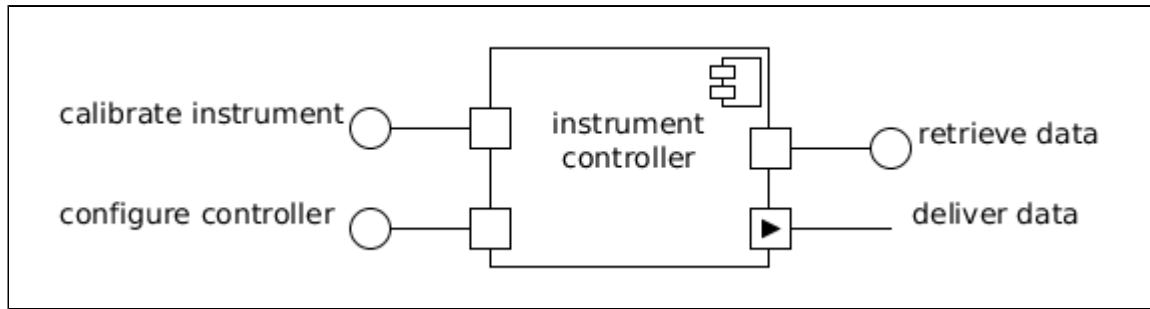
A data store controller should provide two stream interfaces:

- **import data for curation (consumer)** receives data packaged for curation within the associated data store.
- **export curated data (producer)** is used to deliver data stored within the associated data store to another service or resource.

Instrument controller

An integrated raw data source.

An instrument is considered *computationally* to be a source of raw environmental data managed by an acquisition service. An instrument controller object encapsulates the computational functions required to calibrate and acquire data from an instrument.



An instrument controller should provide three operational interfaces:

- **calibrate instrument (server)** provides functions to calibrate the reading of data by an instrument (if possible).
- **configure controller (server)** provides functions to configure how and when an instrument delivers data to a data store.
- **retrieve data (server)** provides functions to directly request data from an instrument.

An instrument controller should provide at least one stream interface:

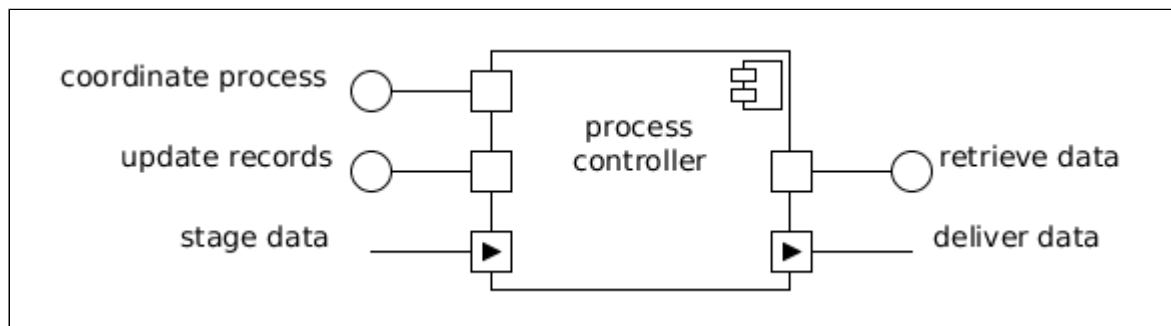
- **deliver raw data (producer)** is used to deliver raw data streams to a designated data store.



'Instrument' is a logical entity, and may to multiple physical entities deployed in the real world should they act in tandem sufficiently closely to justify being treated as one data source. Any instrument represented by an instrument controller should however be considered independently configurable and monitorable from other instruments managed by the same acquisition service.

Process controller

Part of the execution platform that controls the deployment of processing components and the assignment of processing tasks.



A process controller object encapsulates the functions required for using an execution resource (generically, any computing platform that can host some process) as part of any infrastructure workflow.

A process controller should provide at least three operational interfaces:

- **coordinate process (server)** provides functions for controlling the execution resource associated with a given process controller.
- **retrieve data (server)** provides functions for retrieving data from an execution resource.
- **update records (server)** provides functions for modifying data on an execution resource, including preparing the resource for the ingestion of bulk data delivered through its *stage data* stream interface.

A process controller should provide at least two stream interfaces:

- **stage data (consumer)** is used to acquire data sent from the data store objects of a research infrastructure needed as part of some process.
- **deliver dataset (producer)** is used to deliver any new data produced for integration into the data curation store objects of a research infrastructure.

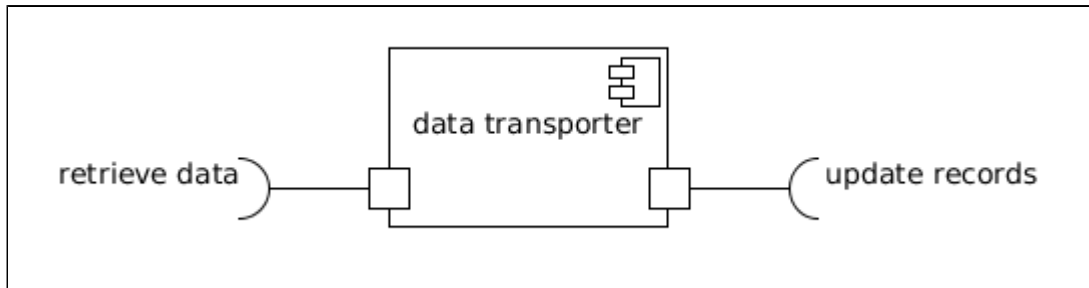
Data transporter

Generic binding object for data transfer interactions.

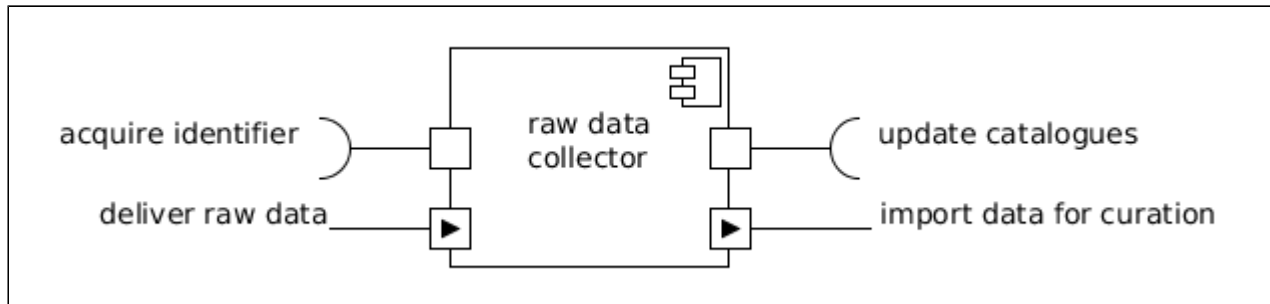
A data transporter binding object encapsulates the coordination logic required to deliver data into and out of the data stores of a RI. A data transporter object is created whenever data is to be streamed from one locale to another.

A data transporter is configured based on the data transfer to be performed, but must have at least the following two interfaces:

- **update records (client)** is used to inform downstream resources about impending data transfers.
- **retrieve data (client)** is used to request data from a given data source.



Raw data collector



Binding object for raw data collection.

A sub-class of **data_transporter** binding object encapsulating the functions required to move and package raw data collected by acquisition objects.

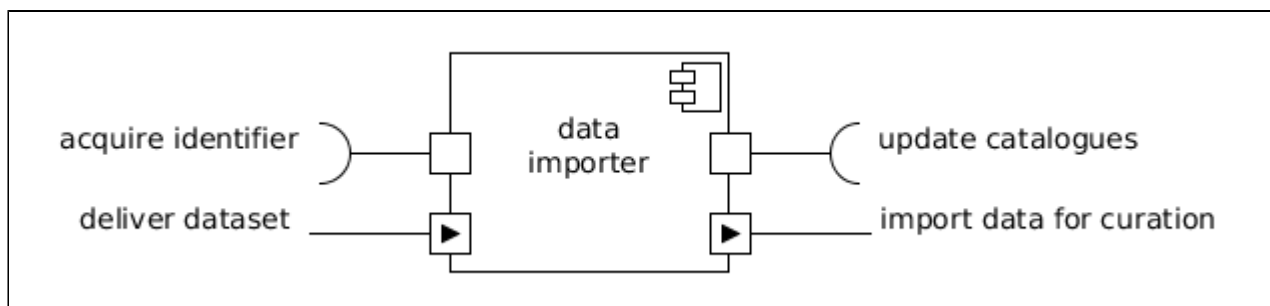
A raw data collector should provide at least two operational interfaces in addition to those provided by any data transporter:

- **acquire identifier (client)** is used to request a new persistent identifier to be associated with the data being transferred. Generally, identifiers are requested when importing new data into an infrastructure.
- **update catalogues (client)** is used to update (or initiate the update of) data catalogues used to describe the data held within an infrastructure to account for new datasets.

A raw data collector must also provide two stream interfaces through which to pass data:

- **deliver raw data (consumer)** is used to collect raw data sent by instruments (data acquisition objects).
- **import data for curation (producer)** is used to deliver (repackaged) raw data to data store objects.

Data importer



Binding object for importing external datasets.

A sub-class of **data_transporter** binding object encapsulating the functions required to move and package external datasets from outside sources into the RI.

A data importer should provide at least two operational interfaces in addition to those provided by any data transporter:

- **acquire identifier (client)** is used to request a new persistent identifier to be associated with the data being transferred. Generally, identifiers are requested when importing new data into an infrastructure.
- **update catalogues (client)** is used to update (or initiate the update of) data catalogues used to describe the data held within an infrastructure to account for new datasets.

A data importer must also provide two stream interfaces through which to pass data:

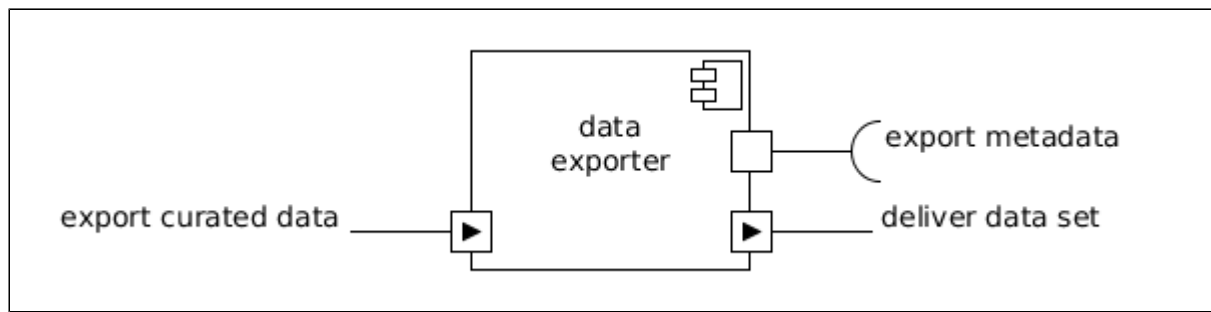
- **deliver dataset (consumer)** is used to retrieve external datasets stored in external data stores outside of the RI.
- **import data for curation (producer)** is used to deliver (repackaged) datasets to one or more data stores within the RI.

Data exporter

Binding object for exporting curated datasets.

A subclass of [data transporter](#)

binding object encapsulating the functions required to move and package curated datasets from the data curation objects to an outside destination.



A data exporter should provide at least one operational interface in addition to those provided by any data transporter:

- **export metadata (client)** is used to retrieve any additional metadata to be associated with the data being transferred.

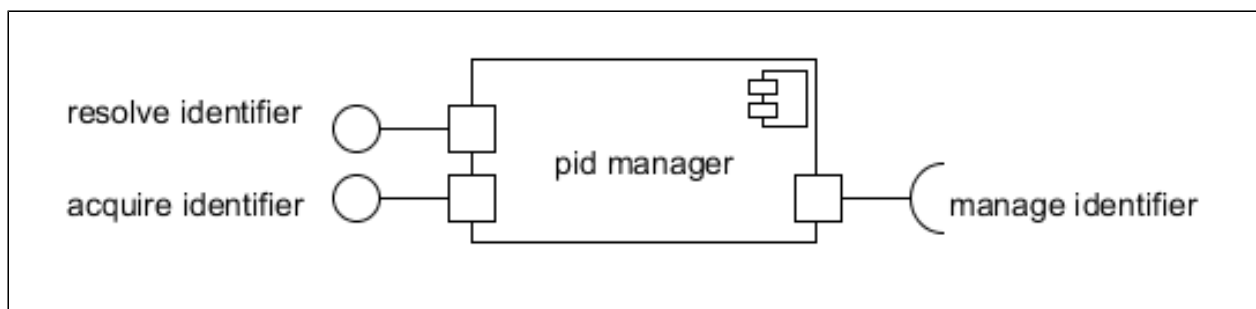
A data exporter must also provide two stream interfaces through which to pass data:

- **export curated data (consumer)** is used to retrieve curated datasets stored within data stores.
- **deliver dataset (producer)** is used to deliver (repackaged) curated data to a designated external data store outside of the RI.

PID manager

Annotation managing

the generation assignment and registration of identifiers.



PID manager object encapsulates the functions required to assign, register and resolve 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 at least two operational interfaces:

- **acquire identifier (server)** provides a persistent identifier for a given entity.
- **resolve identifier (server)** resolves identifiers, interpret and redirect requests to actual data objects.
- **manage identifier (client)** function for retrieving, updating and deleting identifiers interacting with a pid service.