The ENVRI Reference Model (ENVRI RM) version 2.2, 30th October 2017

http:/envri.eu/rm

LEADING BENEFICIARY: CARDIFF UNIVERSITY

Author(s):	Beneficiary/Institution
Abraham Nieva de la Hidalga	Cardiff University, UK
Barbara Magagna	Environment Agency, Austria
Markus Stocker	University of Bremen, Germany
Alex Hardisty	Cardiff University, UK
Paul Martin	University of Amsterdam, Netherlands
Zhiming Zhao	University of Amsterdam, Netherlands
Malcolm Atkinson	University of Edinburgh, UK
Keith Jeffery	NERC, UK

Document type: Snapshot of wiki pages tree, http://envri.eu/rm

Dissemination level: PUBLIC

Date of Production: 15th November 2017

This document is a snapshot of record from multiple pages of the wiki describing the ENVRI Reference Model, version 2.2 30th October 2017. This snapshot document can also be found in the RM wiki itself, as an attachment on the page: “Download of ENVRI Reference Model”.

ENVRI Reference Model

This is the home of the ENVRI Reference Model v2.2, published 30th October 2017 and guidelines on how to use it. Click on the navigation links to the left, or search using the search box above.

This space is under active development. If you find something incorrect or missing, or something that is not clearly explained, please tell us by emailing us at **<envri-rm@list.uva.nl>.**

Contents

Analysis of Common Requirements

The ENVRI Reference Model is originally based on a pre-study of 6 ESFRI Environmental Research Infrastructures (RI), carried out as part of the ENVRI project. It has been updated during the ENVRIplus project from the results of a study of these original 6 and a further 13 RIs. The reports of these studies can be downloaded as follows:

Requirements studies	Notes	Date	Authors	Download
ENVRIplus deliverable D5.1: A consistent characterisation of existing and planned RIs	A version of the study carried out during the ENVRIplus project, with minor editorial corrections beyond the version submitted to the European Commission.	24 May 2016	Malcolm Atkinson (UEDIN) et al.	[**.docx**][**.pdf**]
ENVRI deliverable D3.3: Analysis of Common Requirements For ENVRI Research Infrastructures V1.0 (Final)	A final version report of the study carried out during the ENVRI project, as submitted to the European Commission.	01 May 2013	Yin Chen (CU)	[**.doc**][**.pdf**]

Learning Materials

A new series of courses has been designe to introduce the ENVRI RM. This includes lessons target to RI personal, developers and decision makers. This courses are delivered in the ENVRI community **training platform** deployed and maintained by LifeWatch Italy / University of Salento on behalf of the ENVRI Community.

Available Courses

Practical Introduction to the ENVRI RM

This course is structured in 9 lessons. The course provides a introduction to the main concepts of the ENVRI Reference Model and its 3 logical viewpoints, Science Viewpoint, Information Viewpoint and Computational Viewpoint. The material is present in the context of a real life use case. This is an introductory level course for systems engineers. Access the course on the **ENVRI community training platform**.

Courses Under Development

Practical Introduction to the ENVRI RM - Update

An update of the practical introduction course which will include materials for explaining the use of the Technology and Engineering Viewpoint (4 new lessons)

The Reference Model approach and the value of modelling

This course will contain 10 lessons. The course covers the importance of thinking about becoming involved with the ENVRI RM. This is an introductory level course that would benefit those thinking of becoming familiar with the RM, and those supervising staff in that position.

Previous Training Material

Guideline for Using the Reference Model

Note
The guidelines for using the ENVRI RM are based on version 1.1, some of the information they contain may differ with respect to the current version of the ENVRI RM, however they are still relevant because they present practical examples of using the ENVRI RM for different purposes (they are included in **Appendix D**).

Versions	Notes	Date	Authors	Download
Guideline for Using the Reference Model (Final)	A final version submitted to the European Commission. These are the original guidelines, produced during the ENVRI project. They are still relevant but the status of the RIs they describe may have changed.	30/09/2013	Yin Chen (CU), Barbara Magagna (EAA), Paul Martin (UEDIN), Alex Hardisty(CU), Alun Preece(CU), Herbert Schentz(EAA), Zhiming Zhao(UvA), Robert Huber(UniHB), Ingemar Haggstrom (EISCAT), Ville Savolainen(CSC), Malgozata Krakowian(EGI.eu)	[**.doc**][**.pdf**]

Video Tutorials

Note
These tutorials were created for ENVRI RM version 1.1, some of the information they contain may differ with respect to the current version of the ENVRI RM

- ENVRI Reference Model: an Overview. [**.pptx**]

- Main Processes of the ENVRI Reference Model – Corresponding Viewpoint [**.pptx**]

Publications

- Martin P, Chen Y, Hardisty A, Jeffery K, and Zhao Z. (2017) Computational Challenges in Global Environmental Research Infrastructures. In: Terrestrial Ecosystem Research Infrastructures: Challenges and Opportunities. Eds. Chabbi A, and Loescher HW. 2017 CRC Press ISBN 9781498751315. doi: **10.1201/9781315368252**
- Zhao Z, Martin P, Grosso P, Los W, de Laat C, Vermeulen A, Jeffrey K, Castelli D, Hardisty A, Legre Y, Kutsch W. (2015) Reference Model Guided System Design and Implementation for Interoperable Environmental Research Infrastructures. Presented at: e-Science 2015: IEEE 11th International Conference on e-Science, Munich, Germany, 31 August - 4 September 2015. e-Science (e-Science), 2015 IEEE 11th International Conference on. IEEE, pp. 551-556. doi: 10.1109/eScience.2015.41 Near-final text: [**.pdf**]
- Martin, P., Grosso, P., Magagna, B., Schentz, H., Chen, Y., Hardisty, A., Los, W., Jeffery, K., de Laat, C., Zhao, Z. (2015) Open Information Linking for Environmental Research Infrastructures. Presented at: IEEE 11th International Conference on e-Science, Munich, Germany, 31 August 2015 - 4 September 2015. e-Science (e-Science), 2015 IEEE 11th International Conference on. IEEE, pp. 513-520. doi: 10.1109/eScience.2015.66 Near-final text: [**.pdf**]
- Chen, Y., Martin, P., Schentz, H., Magagna, B., Zhao, Z., Hardisty, A., Preece, A., Atkinson, M., Huber, R. & Legre, Y. (2013), "A Common Reference Model for Environmental Science Research Infrastructures", in the *Proceedings of the 27th Conference on Environmental Informatics* 2013, p665-673, 2013. [**.pdf**]
- Chen, Y., Hardisty, A., Preece, A., Martin, P., Atkinson, M., Zhao, Z., Magagna, B., Schentz, H. & Legre, Y. (2013). "Analysis of Common Requirements for Environmental Science Research Infrastructures", *in the Proceeding of Science (PoS)* SISSA, *PoS(ISGC 2013)032* [**.pdf**]
- Zhao, Z., Grosso, P. & Laat, C. de (2012). "OEIReference Model: An Open Distributed Processing based Interoperability Reference Model for e-Science", *Cloud&Grid interoperability workshop,* Gwangju, Korean, 2012*.*
- Zhao, Z., van der Ham, J., Taal, A., Koning, R., Dumitru, C., Wibisono, A., Grosso, P., de Laat., C. (2012). "Planning data intensive workflows on inter-domain resources using the Network Service Interface (NSI)", *the 7th Workshop on Workflows in Support of Large-Scale Science, in the context of Supercomputing*, Salt Lake City, 2012;
- Zhao, Z., Dumitru, C., Grosso, P. & Laat, C. de (2012). "Network resource control for data intensive applications in heterogeneous infrastructures", *26th IEEE International Parallel and Distributed Processing Symposium*, Shanghai, 2012.
- Jiang, W., Zhao, Z., Grosso, P., de Laat, C., (2013) Dynamic workflow planning on programmable infrastructure, IEEE Network Architecture Storage, China 2013.

Award

- 1 of 3 Lightning talks in the EGI Community Forum 2014, Helsinki, Finland, 19-23 May 2014. [**.pdf**]

Articles, Posters and Presentations

- Nieva de la Hidalga, A., and Hardisty, A. (2016), "How the ENVRI Reference Model helps to design Research Infrastructures", *ENVRIplus Newsletter No.2, May 2016*. [**link**] [**.pdf**]
- Hardisty, A. (2015). "Reference Models: What are they and why do we need them?", *Blog post, 8th July 2015* **https://alexhardisty.wordpress.com/2015/07/08/reference-models-what-are-they-and-why-do-we-need-them/**.
- Chen, Y., Hardisty, A. (2014), "A Common Reference Model for Environmental Research Infrastructures", *iLEAPS newsletter, Special issue, September 2014. page 17-19* [**.pdf**]
- Chen, Y. ""Using the Reference Model in ICOS Research Infrastructure Design Study -- Updates on Science Viewpoint", *ICOS Interim Scientific Advisory Board, Sep 2014.* [**.doc**]
- Chen, Y., B. Magagna, P. Martine (2014), "Using the Reference Model in ICOS Research Infrastructure Design Study", *ICOS Community, Jun 2014*. [**.doc**]
- Chen, Y., (2013), "ENVRI, Common Operations of Environmental Research Infrastructure", *Data Science Symposium 2013*. [**link**]
- Chen, Y., Häggström, I., Mann, I., Heinselman, C., (2013), "EISCAT 3D incoherent scatter radar system", *Data Science Symposium 2013*. [**link**]
- Chen, Y., Häggström, I., Hardisty, A., Sipos, G., Krakowian, M., Ferreira, N. L., Savolainen, V. (2013). "Towards the Big Data Strategies for EISCAT-3D", *EISCAT International Symposium 2013*, Lancaster, the UK, 2013. [**.pdf**]
- Häggström, I., Chen, Y., Hardisty A., Sipos, G., Krakowian, M., Ferreira, N., & Savolainen, V. (2013). "Towards the Big Data Strategies for EISCAT-3D", *Radiovetenskap och Kommunikation 2013: Generation, Real-Time Processing, Transport, Distribution and Management of Large Raw Data Volumes in the Physical Sciences.*11 - 12 November 2013, KVA, Royal Academy of Sciences, Frescati, Stockholm, 2013. [**link**]
- Preece, A. (2013). "The ENVRI Reference Model", Bulding Global Partnerships - RDA Second Plenary Meeting, Washington DC, US, 16-18 Sep 2013. [**Poster**]
- Zhao, Z., Grosso, P., Los, W., de Laat, C., Chen, Y., Hardisty, A., Martin, P., Herbert, S. & Barbara, M., " OEILM: a semantic linking framework for environmental research infrastructures", 9th *IEEE International Conference on eScience 2013*, Beijing, China, 2013. [**.pdf**]
- Zhao, Z., Grosso, P., Los, W., de Laat, C., Chen, Y., Hardisty, A., Martin, P., Herbert, S. & Barbara, M., " OEILM: a semantic linking framework for environmental research Infrastructures", Supercomputing 2013, Dutch exhibition booth. [**.jpg**]
- Zhao, Z., Grosso, P., Los, W., de Laat, C., Chen, Y., Hardisty, A., Martin, P., Herbert, S. & Barbara, M., " OEILM: a semantic linking framework for environmental research infrastructures", Dutch ICT 2013. [**.jpg**]
- Legre, Y. (2013). "Contributions of Environmental Research Infrastructure to GEOSS", *Presentation in GEO European Projects Workshops 2013,* Barcelona, Spain, 2013. [**.ppt**]

ENVRI Flyers

Current Flyers

- ENVRI RM Overview [**.pdf**]
- OIL-E Overview [**.pdf**]

Previous Flyer

> **Note**
> The flyer was created for ENVRI RM Version 1.1, some of the information may not be accurate with respect to the current version of the ENVRI RM

- [**.pdf**] HD
- [**.pdf**] For Professional Printing Service

Change Log for ENVRI RM V2.2

Motivation	Change	Pages
Updates to the SV as a Result of discussions about alignment with OIL-E	Added a definition of objects to include the description of artefacts, actors, and resources. This also includes the description of institution as an example of an Actor.	**SV Object Classification**
	Added a definition of process and step to community behaviours.	**SV Community Behaviours**

	Added a definition for objectives within the definition of communities.	**SV Communities**
	Added a definition for Research Infrastructure as an SV Object within the definition of communities and the inclusion of the term in the glossary.	**SV Communities**, **SV Object Classification**, **Appendix B Terminology** and **GlossaryResearchInfrastructure**
	Revised and corrected label formats in all SV diagrams.	**SV Communities**, **SV Community Behaviours**, **SV Community Roles**
	Added the definition of Research Infrastructure as a special SV Role.	**SV Community Roles Research Infrastructure**
Updates to the IV as a Result of discussions about alignment with OIL-E	Changed the diagram to match the one provided by Barbara, corrected label formats.	**IV Information Objects**
	Added the definition of a new "data" object.	
	Reordered the definitions alphabetically to facilitate inspection.	
	Added the "backed up" state to the set of persistent data states.	
	Changed the name of "data states" to "persistent data states".	
Updates to IV actions derived from alignment with D5.5 (work with Keith)	Added the definition of "semantic annotation" action to distinguish from simple "annotation"	**IV Information Action Types**
	Modified the definition of "annotate data" and "annotate metadata" actions as specialisations of semantic annotation. Additionally added definition of Free Text Annotation as an information action and as glossary term.	
	Changed the diagrams to correct label formats and added the new semantic annotation action.	
Updates to CV objects derived from alignment with D5.5 (work with Keith)	Changed the definition of the "catalogue service" object interfaces to better reflect the operations provided by the service.	**CV Service Objects**
	Changed the "data broker" object interface to align with changes to "catalogue service".	
Update Glossary	Revised the definitions of Metadata Harvester, metadata harvesting, data collector and data collection	**Appendix B Terminology and Glossary**
Updated Glossary (Midterm review request, work with Zhimin, Paul, Markus and Barbara)	Revised and edited defintions included in the RM glossary	**Appendix B Terminology and Glossary**
Engineering Viewpoint	Proposal for the definition of the Engineering Viewpoint, derived from discussions with architecture team and RI visits	**Engineering Viewpoint (Draft)**
Technology Viewpoint	Proposal for the definition of the Technology Viewpoint, derived from discussions with architecture team and RI visits	**Technology Viewpoint (Draft)**

Recently Updated

Template for requirements
about 2 hours ago • updated by Barbara Magagna • view change

EARS_INTEL_TheEasyApproachToRequirementsSyntax.pdf
about 2 hours ago • attached by Barbara Magagna

volere-template.pdf
about 2 hours ago • attached by Barbara Magagna

questions_V6.doc
about 2 hours ago • attached by Barbara Magagna

EV Architectural Model (Draft)
about 2 hours ago • updated by Abraham Nieva de la Hidalga • view change

EVReferenceArchitecture.png
about 2 hours ago • attached by Abraham Nieva de la Hidalga

Provenance actions
about 2 hours ago • updated by Barbara Magagna • view change

Provenance service
about 2 hours ago • updated by Barbara Magagna • view change

Catalogue Discovery Object (Draft)
Nov 03, 2017 • updated by Abraham Nieva de la Hidalga • view change

The ENVRI and ENVRIplus Projects
Nov 03, 2017 • updated by Abraham Nieva de la Hidalga • view change

ENVRI Reference Model
Nov 03, 2017 • updated by Abraham Nieva de la Hidalga • view change

SV Community Roles
Nov 01, 2017 • updated by Abraham Nieva de la Hidalga • view change

Object Configuration (Draft)
Nov 01, 2017 • updated by Abraham Nieva de la Hidalga • view change

Technology Viewpoint (Draft)
Nov 01, 2017 • updated by Abraham Nieva de la Hidalga • view change

Appendix B Terminology and Glossary
Nov 01, 2017 • updated by Abraham Nieva de la Hidalga • view change

Download of ENVRI Reference Model

The ENVRI Reference Model is a work-in-progress, developed by the ENVRI and ENVRIplus projects, intended for interested parties to directly comment on and contribute to.

From time to time new versions of the document will be released which are the snapshots of development milestones.

Versions	Notes	Date	Authors	Download
ENVRI Reference Model V2.2	Version 2.2, incorporating further changes and detail arising from ENVRIplus requirements, alignment with OIL-E, and Architecture proposal (see the **Change Log for ENVRI RMV2.2**).	30 Oct 2017	Abraham Nieva de la Hidalga (CU), Barbara Magagna (EAA), Markus Stocker (UniHB), Alex Hardisty (CU), Paul Martin (UvA), Zhiming Zhao (UvA), Malcolm Atkinson (UEDIN), Keith Jeffery (NERC)	[**.pdf**]
ENVRI Reference Model V2.1	Version 2.1, incorporating further changes arising from ENVRIplus requirements analysis and the new larger community of Environmental Research Infrastructures represented in the ENVRIplus project. Full details of all changes are documented in **deliverable D5.2**.	09 Nov 2016	Abraham Nieva de la Hidalga (CU), Barbara Magagna (EAA), Markus Stocker (UniHB), Alex Hardisty (CU), Paul Martin (UvA), Zhiming Zhao (UvA), Malcolm Atkinson (UEDIN)	[**.pdf**]
ENVRI Reference Model V2.0	Version 2.0, incorporating changes arising from ENVRIplus requirements analysis and realignment along lines of data lifecycle model	27 July 2016	Abraham Nieva de la Hidalga (CU), Barbara Magagna (EAA), Markus Stocker (UniHB), Alex Hardisty (CU), Paul Martin (UvA), Zhiming Zhao (UvA), Malcolm Atkinson (UEDIN)	[**.pdf**]

ENVRI Reference Model V1.1	The second major version	30 Aug 2013	Yin Chen (CU), Paul Martin (UEDIN), Herbert Schentz (EAA), Barbara Magagna (EAA), Zhiming Zhao (UvA), Alex Hardisty (CU), Alun Preece (CU), Malcolm Atkinson (UEDIN)	[.doc][.pdf]
ENVRI Reference Model V1.0 (Final)	A final version ready to submit to the European Commission	01 May 2013	Yin Chen (CU), Paul Martin (UEDIN), Herbert Schentz (EAA), Barbara Magagna (EAA), Zhiming Zhao (UvA)	[.doc][.pdf]
ENVRI Reference Model V1.0 (Draft)	A draft version ready for Internal review	01 Apr 2013	**Yin Chen** (CU), **Paul Martin** (UEDIN), **Herbert Schentz** (EAA), **Barbara Magagna** (EAA), **Zhiming Zhao** (UvA), Alex Hardisty (CU), Alun Preece (CU), Malcolm Atkinson (UEDIN)	[.doc][.pdf]

Getting started with the ENVRI RM

The ENVRI Reference Model (ENVRI RM, RM) exists to illustrate common characteristics of environmental science research infrastructures in order to provide a common language and understanding, promote technology and solution sharing and improve interoperability.

About

Independent development of research infrastructures leads to unnecessary replication of technologies and solutions whilst the lack of standard definitions makes it difficult to relate experiences in one infrastructure with those of others. The ENVRI Reference Model (ENVRI RM) uses Open Distributed Processing (ODP) in order to model the "archetypical" environmental research infrastructure. The use of the ENVRI RM to illustrate common characteristics of existing and planned European Environmental Research Infrastructures from a number of different perspectives provides a common language for and understanding of those infrastructures, promotes technology and solution sharing between infrastructures, and improves interoperability between implemented services.

Intended Audience

The intended audience of this document is the **ENVRI community** as well as other organisations or individuals that are interested in understanding the top level conceptual architecture that underpins the construction of such research infrastructures. In particular, the intended primary audience the Reference Model includes [33]:

- Research Infrastructures Implementation teams:
  - Architects, designers, and integrators;
  - Engineers – to enable them to be able to drill down directly to find required knowledge;
- Research Infrastructure Operations teams; and
- Third party solution or component providers.

The Reference Model is also intended for research infrastructure leaders and service centre staffs.

The Reference Model can be read by others who want to better understand the ENVRI community work, to gain understanding necessary to make contributions to the standardisation processes of environmental research infrastructures.

Document Structure

Introduction introduces the motivation and background knowledge of the ENVRI RM.

Model Overview presents an overview of the ENVRI RM against the backdrop of a typical lifecycle for research data.

The ENVRI Reference Model is a detailed description of the ENVRI RM from the Open Distributed (ODP) Viewpoints perspectives.

Conclusions and Future Work concludes this work.

Appendices are not formally part of the reference model. They provide additional information that may be helpful and for the convenience of the reader.

Appendix A Common Requirements of Environmental Research Infrastructures presents the full list of the required functionalities that is the result of the investigations of the common requirements of Research Infrastructures.

Appendix B Terminology and Glossary is a glossary of terms, and consists of concepts and terms defined throughout the ENVRI RM.

How to Read

- The primary audience of the ENVRI RM should generally read the whole documentation, starting with the **Introduction** and **Model Overview**. Such readers then should proceed to the **Science Viewpoint** and the **Information Viewpoint** before looking at the **Computational Viewpoint**. It is not necessary to read everything nor to read in order. The tutorials given below are useful entry points. Elsewhere (link to be provided) we give detailed guidance on how best to engage with the Reference Model for different purposes.
- The leaders of research infrastructures, and service centre staff may want to read the introduction and background knowledge in **Introduction** and **Model Overview**.
- Readers who have general interests in the ENVRI RM may want to read **Introduction**.

Learning

The ENVRIplus project is developing custom materials for different types of users of the ENVRI RM which complement this wiki and the other dissemination materials (**Learning Materials**)

The ENVRI and ENVRIplus Projects

Frontier environmental research increasingly depends on a wide range of data and advanced capabilities to process and analyse them. The original ENVRI project, "Common Operations of Environmental Research infrastructures" (2011 - 2014) was a collaboration in the **ESFRI** Environment Cluster, with support from ICT experts, to develop common e-science components and services for their facilities. The results are intended to speed up the construction of these Environmental Sciences research infrastructures and to allow scientists to use the data and software from each facility to enable multi-disciplinary science. The work is continuing (2015 - 2019) as part of the **ENVRIplus project "Environmental Research Infrastructures Providing Shared Solutions for Science and Society"**.

The focus is on developing common capabilities including software and services for environmental and e-infrastructure communities. While the Environmental Sciences research infrastructures are very diverse, they face common challenges including data capture from distributed sensors, metadata standardisation, management of high volume data, workflow execution and data visualisation. Common standards, deployable services and tools will be adopted by each infrastructure as it progresses into its construction phase.

The ENVRI and ENVRIplus projects deliver a common reference model, the "ENVRI Reference Model" or "ENVRI RM" created by capturing the functional and other capabilities of each ESFRI-ENV infrastructure. This model and the development driven by the testbed deployments result in ready-to-use systems that can be integrated into the environmental research infrastructures.

The projects put emphasis on synergy between advanced developments, not only among the infrastructure facilities, but also with ICT providers and related e-science initiatives. These links will facilitate system deployment and the training of future researchers, and ensure that the inter-disciplinary capabilities established here remain sustainable beyond the lifetime of the project.

Introduction

Purpose and Scope

All research infrastructures for environmental sciences (the so-called 'ENVRIs') although very diverse, have some common characteristics, enabling them potentially to achieve a greater level of interoperability through the use of common standards and approaches for various functions. The objective of the ENVRI Reference Model is to develop a common framework and specification for the description and characterisation of computational and storage infrastructures. This framework can support the ENVRIs to achieve seamless interoperability between the heterogeneous resources of their different infrastructures.

The ENVRI Reference Model serves the following purposes [**1**]:

- to provide a way for structuring thinking that helps the community to reach a common vision;
- to provide a common language that can be used to communicate concepts concisely;
- to help discover existing solutions to common problems;
- to provide a framework into which different functional components of research infrastructures can be placed, in order to draw comparisons and identify missing functionality.

The present wiki / document describes the ENVRI Reference Model which:

- captures computational characteristics of data and operations that are common in ENVRI Research Infrastructures; and
- establishes a taxonomy of terms, concepts and definitions to be used by the ENVRI community.

The Reference Model provides an abstract logical conceptual model. It does not impose a specific architecture. Nor does it impose specific design decisions or constraints on the design of an infrastructure.

The *initial* model (versions 1.0 and 1.1) focused on the urgent and important issues prioritised for ENV research infrastructures including data preservation, data discovery and access, and data publication. It defines a minimal set of functionalities to support these requirements. The *i nitial* model does not cover engineering mechanisms or the applicability of existing standards or technologies.

Version 2.x of the model incrementally extends these core functionalities:

- Version 2.0 is a simplification of the way the Reference Model is presented, to make it easier to understand and become familiar with. Version 2.0 explicitly aligns the RM with a lifecycle oriented view of research data management.

Rationale

Environmental issues will dominate the 21st century [2]. Research infrastructures that provide advanced capabilities for data sharing, processing and analysis enable excellent research and play an ever-increasing role in the environmental sciences as well as in solving societal challenges. The **ENVRIplus project** and its predecessor ENVRI project gathers many of the EU ESFRI and other environmental infrastructures (**ICOS**, **EURO-Argo**, **EISCAT-3D**, **LifeWatch**, **EPOS**, **EMSO**, etc.) to find common solutions to common problems, including use of common software solutions. The results, including the ENVRI Reference Model will accelerate the construction of these infrastructures and improve interoperability among them. The experiences gained will also benefit building of other advanced research infrastructures.

The primary objective of ENVRI is to agree on a reference model for joint operations. This will enable greater understanding and cooperation between infrastructures since fundamentally the model will serve to provide a universal reference framework for discussing many common technical challenges facing all of the ESFRI-ENV infrastructures. By drawing analogies between the reference components of the model and the actual elements of the infrastructures (or their proposed designs) as they exist now, various gaps and points of overlap can be identified [3].

The ENVRI Reference Model is based on the design experiences of the state-of-the-art environmental research infrastructures, with a view of informing future implementation. It tackles multiple challenging issues encountered by existing initiatives, such as data streaming and storage management; data discovery and access to distributed data archives; linked computational, network and storage infrastructure; data curation, data integration, harmonisation and publication; data mining and visualisation, and scientific workflow management and execution. It uses Open Distributed Processing (ODP), a standard framework for distributed system specification, to describe the model.

To our best knowledge there is no existing reference model for environmental science research infrastructures. This work intends to make a first attempt, which can serve as a basis to inspire future research explorations.

There is an urgent need to create such a model, as we are at the beginning of a new era. The advances in automation, communication, sensing and computation enable experimental scientific processes to generate data and digital objects at unprecedentedly great speeds and volumes. Many infrastructures are starting to be built to exploit the growing wealth of scientific data and enable multi-disciplinary knowledge sharing. In the case of ENVRI, most investigated RIs are in their planning / construction phase. The high cost attached to the construction of environmental infrastructures require cooperation on the sharing of experiences and technologies, solving crucial common e-science issues and challenges together. Only by adopting a good reference model can the community secure interoperability between infrastructures, enable reuse, share resources and experiences, and avoid unnecessary duplication of effort.

The contribution of this work is threefold:

- The model captures the computational requirements and the state-of-the-art design experiences of a collection of representative research infrastructures for environmental sciences. It is the first reference model of this kind which can be used as a basis to inspire future research.
- It provides a common language for communication to unify understanding. It serves as a community standard to secure interoperability.
- It can be used as a base to drive design and implementation. Common services can be provided which can be widely applicable to various environmental research infrastructures and beyond.

Basis

The ENVRI Reference Model is built on top of the Open Distributed Processing (ODP) framework [4, 5, 6, 7]. ODP is an international standard for architecting open, distributed processing systems. It provides an overall conceptual framework for building distributed systems in an incremental manner.

The reasons for adopting the ODP framework in the ENVRI project come from three aspects:

- It enables large collaborative design activities;
- It provides a framework for specifying and building large or complex system that consists of a set of guiding concepts and terminology. This provides a way of thinking about architectural issues in terms of fundamental patterns or organising principles; and
- Being an international standard, ODP offers authority and stability.

ODP adopts the ***object modelling*** approach to system specification. ISO/IEC 10746-2 [5] includes the formal definitions of the concepts and terminology adopted from object models, which serves as the foundation for expressing the architecture of ODP systems. The modelling concepts fall into three categories [4, 5]:

- Basic modelling concepts for a general object-based model;
- Specification concepts to allow designers to describe and reason about ODP system specifications;
- Structuring concepts, including organisation, the properties of systems and objects, management, that correspond to notions and structures that are generally applicable in the design and description of distributed systems.

ODP is best known for its use of viewpoints. A *viewpoint* (on a system) is an abstraction that yields a specification of the whole system related to a particular set of concerns. The ODP reference model defines five specific viewpoints as follows [4, 6]:

- The ***Enterprise Viewpoint***, which concerns the organisational situation in which business (research activity in the current case) is to take place; For better communication with ENVRI community, in this document, we rename it as ***Science Viewpoint***.
- The ***Information Viewpoint***, which concerns modelling of the shared information manipulated within the system of interest;
- The ***Computational Viewpoint***, which concerns the design of the analytical, modelling and simulation processes and applications provided by the system;
- The ***Engineering Viewpoint***, which tackles the problems of diversity in infrastructure provision; it gives the prescriptions for supporting the necessary abstract computational interactions in a range of different concrete situations;
- The ***Technology Viewpoint***, which concerns real-world constraints (such as restrictions on the facilities and technologies available to implement the system) applied to the existing computing platforms on which the computational processes must execute.

This version of the ENVRI Reference Model covers 3 ODP viewpoints: the science, information, and computational viewpoints.

Approaches

The approach leading to the creation of the ENVRI Reference Model is based on the analysis of the requirements of a collection of representative environmental research infrastructures, which are reported in two ENVRI deliverables:

- D3.2: Assessment of the State of the Art
- **D3.3**

The ODP standard is used as the modelling and specification framework, which enables the designers from different organisations to work independently and collaboratively. The development starts from a core model and will be incrementally extended based on the community common requirements and interests. The reference model will be evaluated by examining the feasibilities in implementations, and the refinement of the model will be based on community feedback.

Conformance

A conforming environmental research infrastructure should support the common functionalities described in **Model Overview** and the functional and information model described in **The ENVRI Reference Model**.

The ENVRI Reference Model does not define or require any particular method of implementation of these concepts. It is assumed that implementers will use this reference model as a guide while developing a specific implementation to provide identified services and content. A conforming environmental research infrastructure may provide additional services to users beyond those minimally required functions defined in this document.

Any descriptive (or prescriptive) documents that claim to be conformant to the ENVRI Reference Model should use the terms and concepts defined herein in a similar way.

Related Work

Related Concepts

A **reference model** is an abstract framework for understanding significant relationships among the entities of some environment. It consists of a minimal set of unifying concepts, axioms and relationships within a particular problem domain [**8**].

A reference model is not a reference architecture. A **reference architecture** is an architectural design pattern indicating an abstract solution that implements the concepts and relationships identified in the reference model [**8**]. Different from a reference architecture, a reference model is independent from specific standards, technologies, implementations or other concrete details. A reference model can drive the development of a reference architecture or more than one of them [**9**].

It could be argued that a reference model is, at its core, an **ontology**. Conventional reference models, e.g., OSI [**10**], RM-ODP [**4**], OAIS[**11**], are built upon modelling disciplines. Many recent works, such as the DL.org Digital Library Reference Model [**9**], are more ontology-like.

Both models and ontologies are technologies for information representation, but have been developed separately in different domains [**13**]. Modelling approaches have risen to prominence in the software engineering domain over the last ten to fifteen years [**12**]. Traditionally, software engineers have taken very pragmatic approaches to data representation, encoding only the information needed to solve the problem in hand, usually in the form of language, data structures, or database tables. Modelling approaches are meant to increase the productivity by maximising compatibility between systems (by reuse of standardised models), simplifying the process of design (by models of recurring design patterns in the application domain), and promoting communication between individuals and teams working on the system (by a standardisation of the terminology and the best practices used in the application domain) [**13**]. On the other hand, ontologies have been developed by the Artificial Intelligence community since the 1980s. An ontology is a structuring framework for organising information. It renders shared vocabulary and taxonomies which models a domain with the definition of objects and concepts and their properties and relations. These ideas have been heavily drawn upon in the notion of the Semantic Web [**13**].

Traditional views tend to distinguish the two technologies. The main points of argument include but are not limited to:

1. Models usually focus on realisation issues (e.g., the Object-Oriented Modelling approach), while ontologies usually focus on capturing abstract domain concepts and their relationship [**14**].
2. Ontologies are normally used for run-time knowledge exploitation (e.g., for knowledge discovery in a **knowledge base**), but models normally do not [**15**].
3. Ontologies can support reasoning while models cannot (or do not) [**13**].
4. Finally, models are often based on the Closed World Assumption while ontologies are based on the Open World Assumption [**13**].

However, these separations between the two technologies are rapidly disappearing in recent developments. Study [**13**] shows that ‘all ontologies are models’, and ‘almost all models used in modern software engineering qualify as ontologies.’ As evidenced by the growing number of research workshops dealing with the overlap of the two disciplines (e.g., SEKE [16], VORTE [17] , MDSW [18] , SWESE [19] , ONTOSE [20] , WoMM [21]), there has been considerable interests in the integration of software engineering and artificial intelligence technologies in both research and practical software engineering projects [13].

We tend to take this point of view and regard the ENVRI Reference Model as both a model and an ontology. The important consequence is that we can explore further in both directions, e.g., the reference model can be expressed using a modelling language, such as UML (UML4ODP). It can then be built into a tool chain, e.g., to plugin to an integrated development environment such as Eclipse, which makes it possible to reuse many existing UML code and software. On the other hand, the reference model can also be expressed using an ontology language such as RDF or OWL which can then be used in a **knowledge base**. In this document we explore principally from model aspects. In another ENVRI task, T3.4, the ontological aspect of the reference model will be exploited.

Finally, a reference model is a **standard**. Created by ISO in 1970, OSI is probably among the earliest reference models, which defines the well-known 7-layered network communication. As one of the ISO standard types, the reference model normally describes the overall requirements for standardisation and the fundamental principles that apply in implementation. It often serves as a framework for more specific standards [22]. This type of standard has been rapidly adopted, and many reference models exist today, which can be grouped into 3 categories, based on the type of agreement and the number of people, organisations or countries who were involved in making the agreement:

- **Committee reference model** – a widely-based group of experts nominated by organizations who have an interest in the content and application of the standard build the standard.
- **Consensus reference model** – the principle that the content of the standard is decided by general agreement of as many as possible of the committee members, rather than by majority voting. The ENVRI Reference Model falls into this group.
- **Consultation reference model** – making a draft available for scrutiny and comment to anyone who might be interested in it.

Some examples from each of the categories are discussed below, with emphasis on approaches of building the model and technologies the model captures.

Related Reference Models

Committee Reference Models

In this category, we look at those defined by international organisations , such as the Advancing Open Standards for the Information Society (OASIS), the Consultative Committee for Space Data Systems (CCSDS), and the Open Geospatial Consortium (OGC).

The Open Archival Information System (OAIS) Reference Model [11] is an international standard created by CCSDS and ISO which provides a framework, including terminology and concepts for archival concept needed for Long-Term digital information preservation and access.

The OASIS Reference Model for Service Oriented Architecture (SOA-RM) [8] defines the essence of service oriented architecture emerging with a vocabulary and a common understanding of SOA. It provides a normative reference that remains relevant to SOA as an abstract model, irrespective of the various and inevitable technology evolutions that will influence SOA deployment.

The OGC Reference Model (ORM) [23], describes the OGC Standards Baseline, and the current state of the work of the OGC. It provides an overview of the results of extensive development by OGC Member Organisations and individuals. Based on RM-ODP's 5 viewpoints, ORM captures business requirements and processes, geospatial information and services, reusable patterns for deployment, and provides a guide for implementations.

The Reference Model for the ORCHESTRA Architecture (RM-OA) [24] is another OGC standard. The goal of the integrated project ORCHESTRA (Open Architecture and Spatial Data Infrastructure for Risk Management) is the design and implementation of an open, service-oriented software architecture to overcome the interoperability problems in the domain of multi-risk management. The development approach of RM-OA is standard-based which is built on the integration of various international standards. Also using RM-ODP standard as the specification framework, RM-OA describes a platform neutral (abstract) model consisting of the informational and functional aspects of service networks combining architectural and service specification defined by ISO, OGC, W3C, and OASIS [24].

There are no reference model standards yet for environmental science research infrastructures.

Consensus Reference Models

In this category, we discuss those created by non-formal standard organisations.

The LifeWatch Reference Model [25], developed by the EU LifeWatch consortium, is a specialisation of the RM-OA standard which provides the guidelines for the specification and implementation of a biodiversity research infrastructure. Inherited from RM-OA, the reference model uses the ODP standard as the specification framework.

The Digital Library Reference Model [9] developed by DL.org consortium introduces the main notations characterising the whole digital library domain, in particular, it defines 3 different types of systems: (1) Digital Library, (2) Digital Library System, and (3) Digital Library Management System; 7 core concepts characterising the digital library universe: (1) Organisation, (2) Content, (3) Functionality, (4) User, (5) Policy, (6) Quality, and (7) Architecture; and 3 categories of actors: (1) DL End-Users (including, Content Creators, Content Consumers, and Digital Librarians), (2) DL Managers (including, DL Designer, and DL System Administrators), and (3) DL Software Developers.

The Workflow Reference Model [26] provides a common framework for workflow management systems, identifying their characteristics, terminology and components. The development of the model is based on the analysis of various workflow products in the market. The workflow Reference Model firstly introduces a top level architecture and various interfaces it has which may be used to support interoperability between different system components and integration with other major IT infrastructure components. This maps to the ODP Computational Viewpoint. In the second part, it provides an overview of the workflow application program interface, comments on the necessary protocol support for open interworking and discusses the principles of conformance to the specifications. This maps to the ODP Technology Viewpoint.

The Agent System Reference Model [27] provides a technical recommendation for developing agent systems, which captures the features, functions and data elements in the set of existing agent frameworks. Different from conventional methods, a reverse engineering method has been used to develop the reference model, which starts by identifying or creating an implementation-specific design of the abstracted system; secondly, identifying software modules and grouping them into the concepts and components; and finally, capturing the essence of the abstracted system via concepts and components.

Consultation Reference Models

The Data State Reference Model [28] provides an operator interaction framework for visualisation systems. It breaks the visualisation pipeline (from data to view) into 4 data stages (Value, Analytical Abstraction, Visualisation Abstraction, and View), and 3 types of transforming operations (Data Transformation, Visualisation Transformation and Visual Mapping Transformation). Using the data state model, the study [29] analyses 10 existing visualisation techniques including, 1) scientific visualisations, 2) GIS, 3) 2D, 4) multi-dimensional plots, 5) trees, 6) network, 7) web visualisation, 8) text, 9) information landscapes and spaces, and 10) visualisation spread sheets. The analysis results in a taxonomy of existing information visualisation techniques which help to improve the understanding of the design space of visualisation techniques.

The Munich Reference Model [30] is created for adaptive hypermedia applications which is a set of nodes and links that allows one to navigate through the hypermedia structure and that dynamically “adapts” (personalise) various visible aspects of the system to individual user’s needs. The Munich Reference Model uses an object-oriented formalisation and a graphical representation. It is built on top of the Dexter Model layered structure, and extends the functionality of each layer to include the user modelling and adaptation aspects. The model is visually represented using in UML notation and is formally specified in Object Constraint Language (which is part of the UML).

While these works use a similar approach to the development of the reference model as the ENVRI-RM, which is based on the analysis of existing systems and abstracts to obtain the ‘essence’ of those systems, a major difference is that these works have not normally met with significant feedback or been formally approved by an existing community, with the consequence that they express less authority as a standard.

Other Related Standards

Data Distribution Service for Real-Time Systems (DDS) [31], an Object Management Group (OMG) standard, is created to enable scalable, real-time, dependable, high performance, interoperable data exchanges between publishers and subscribers. DDS defines a high-level conceptual model as well as a platform-specific model. UML notations are used for specification. While DDS and the ENVRI share many similar views in design and modelling, DDS focuses on only one specific issue, i.e., to model the communication patterns for real-time applications; while ENVRI aims to capture a overall picture of requirements for environmental research infrastructures.

Published by the web standards consortium OASIS in 2010, the Content Management Interoperability Services (CMIS) [32] is an open standard that allows different content management systems to inter-operate over the Internet. Specially, CMIS defines an abstraction layer for controlling diverse document management systems and repositories using web protocols. It defines a domain model plus web services and Restful AtomPub bindings that can be used by applications to work with one or more Content Management repositories/systems. However as many other OASIS standards, CMIS is not a conceptual model and is highly technology dependent [32].

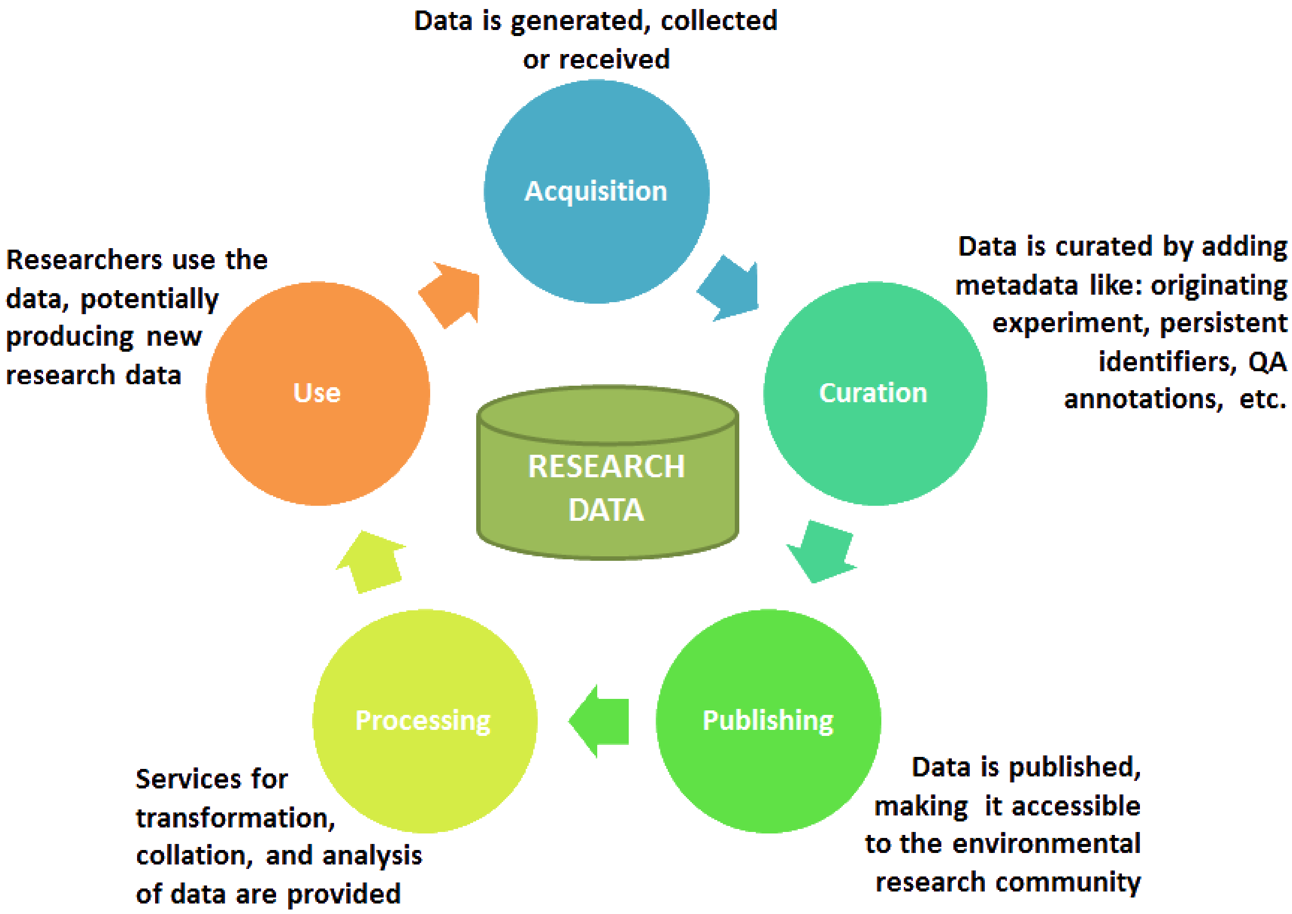
Model Overview

The Research Data Lifecycle within Environmental Research Infrastructures

The ENVRI and ENVRIplus project investigated a collection of more than 20 representative environmental research infrastructures (RIs) from different areas. By examining these research infrastructures and their characteristics, a common data lifecycle was identified. The data lifecycle is structured in five phases: *Data Acquisition*, *Data Curation*, *Data Publishing*, *Data Processing* and *Data Use*
The fundamental reason of the division of the data lifecycle is based on the observation that all applications, services and software tools are designed and implemented around five major activities: acquiring data, storing and preserving data, making the data publicly available, providing services for further data processing, and using the data to derive different data products. This data lifecycle is fairly general and all research infrastructures investigated exhibit behaviour that aligns with its phases. Consequently, the ENVRI-RM is structured in line with the five phases of the data life-cycle.

This lifecycle begins with the acquisition of data from a network of integrated data collecting entities (seismographs, weather stations, robotic buoys, human observers, or simulators) which is then registered and curated within a number of data stores belonging to an infrastructure or one of its delegate infrastructures. This data is then made accessible to parties external to the infrastructure, as well as to services within the infrastructure. This results in a natural partitioning into data acquisition, curation and publishing. In addition, RIs may provide services for processing data, the results of this processing can then produce new data to be stored within the infrastructure. Finally, the broader research community outside of the RI can design experiments and analyses on the published data and produce new data, which in turn can be passed to the same RI or to other RI for curation, publishing and processing, restarting the lifecycle.

The activities of each research infrastructure can align with this lifecycle. However, research infrastructures will tend to optimise and concentrate more on some phases. For instance, some research infrastructures concentrate mostly on the acquisition of data, while others focus their expertise on curation or publishing. ENVRI RM assumes that the research infrastructures can complement and integrate with each other to support the entire data lifecycle. Integration is achieved through providing a set of capabilities via interfaces invoked within systems (or subsystems) which can be used within the infrastructures but also across boundaries. In the ENVRI RM, an interface is an abstraction of the behaviour of an object that consists of a subset of the interactions expected of that object together with the constraints imposed on their occurrence.


Data is generated, collected or received
Acquisition
Data is curated by adding metadata like: originating experiment, persistent identifiers, QA annotations, etc.
Curation
RESEARCH DATA
Publishing
Data is published, making it accessible to the environmental research community
Processing
Services for transformation, collation, and analysis of data are provided
Use
Researchers use the data, potentially producing new research data

The Research Data Lifecycle

Data Acquisition

*In the **data acquisition phase** the research infrastructure collects raw data from registered sources to be stored and made accessible within the infrastructure.*

The data acquisition phase supports collecting raw data from sensor arrays and other instruments, as well as from human observers, and brings those data into the data management part (ie., ICT sub-systems) of the research infrastructure. Within the ENVRI-RM, the acquisition phase is considered to begin upon point of data entry into the RI systems. The acquisition phase as modeled in the ENVRI RM starts from the design of the experiment. Acquisition is typically distributed across networks of observatories and stations. The data acquired is generally assumed to be non-reproducible, being associated with a specific (possibly continuous) event in time and place. As such, the assignment of provenance (particularly data source and timestamp) is essential. Real-time data streams may be temporarily stored, sampled, filtered and processed (e.g., based on applied quality control criteria) before being ready for curation. Control software is often deployed to manage and schedule the execution and monitoring of data flows. Data collected during the acquisition phase ultimately enters the data curation phase for preservation, usually within a specific time period.

Data Curation

*In the **data curation phase** the research infrastructure stores, manages and ensures access to all persistent data-sets produced within the infrastructure.*

The data curation phase facilitates quality control and preservation of scientific data. The data curation functionalities are typically implemented across one or more dedicated data centres. Data handled at this phase include raw data products, metadata and processed data. Where possible, processed data should be reproducible by executing the same process on the same source data-sets, supported by provenance data. Operations such as data quality verification, identification, annotation, cataloguing, replication and archival are often provided. Access to curated data from outside the infrastructure is brokered through independent data access mechanisms. There is usually an emphasis on non-functional requirements for data curation satisfying availability, reliability, utility, throughput, responsiveness, security and scalability criteria.

Data Publishing

In the ***data publishing phase*** *the research infrastructure enables discovery and retrieval of scientific data to internal and external parties.*

The data publishing phase enables discovery and retrieval of data housed in data resources managed as part of data curation. Data publishing often provide mechanisms for presenting or delivering data products. Query and search tools allow users or upstream services to discover data based on metadata or semantic linkages. Data handled during publishing need not be homogeneous. When supporting heterogeneous data, different types of data (often pulled from a variety of distributed data resources) can be converted into uniform representations with uniform semantics resolved by a data discovery service. Services for harvesting, compressing and packaging data and metadata, as well as encoding services for secure transfer can be provided. Data publishing is controlled using rights management, authentication, and authorisation policies.

Data Processing

In the ***data processing phase*** *the research infrastructure provides a toolbox of services for performing a variety of data processing tasks. The scope of data processing is very wide.*

The data processing phase enables the aggregation of data from various sources, as well as conduct of experiments and analyses upon that data. During this phase data tends to be manipulated, leading to both/either derived and/or recombined data. To support data processing, a research infrastructure is likely to offer service operations for statistical analysis and data mining, as well as facilities for carrying out scientific experiments, modelling and simulation, and visualisation. Performance requirements for processing scientific data during this phase tend to be concerned with scalability, which can be addressed at the level of engineering and technical solutions to be considered (e.g., by making use of Cloud computing services). The data products generated during processing may themselves be curated and preserver within the RI.

Data Use

In the ***data use phase*** *the research infrastructure supports users of an infrastructure in gaining access to data and facilitating the preservation of derived data products.*

The data use phase provides functionalities that manage and track users' activities while supporting the users to conduct their research activities which may result in the creation of new data products. Data 'handled' and produced at this phase are typically user-generated data and communications. The data use phase requires supporting activities such as interactive visualisation, standardised authentication, authorisation and accounting protocols, and the use of virtual organisations. This is the most advanced form of data processing, at this phase the research infrastructure implements an interface with the wider world in which it exists.

Lifecycle Support Inter- and Intra- Research Infrastructure Relationships

Each research infrastructure supports the data lifecycle to a different degree. According to the scope of a particular research infrastructure, some core activities align strongly with some of the phases while other phases are not so comprehensively supported. In this case, the integration of the research infrastructures and their external supporting systems and services help in the overall fulfilment of the research data lifecycle. For these cases, the major integration points are those at the transition between phases of the data lifecycle. These integration points are important to build the internal subsystems of the research infrastructure, as well as to integrate the research infrastructure with other research infrastructures.

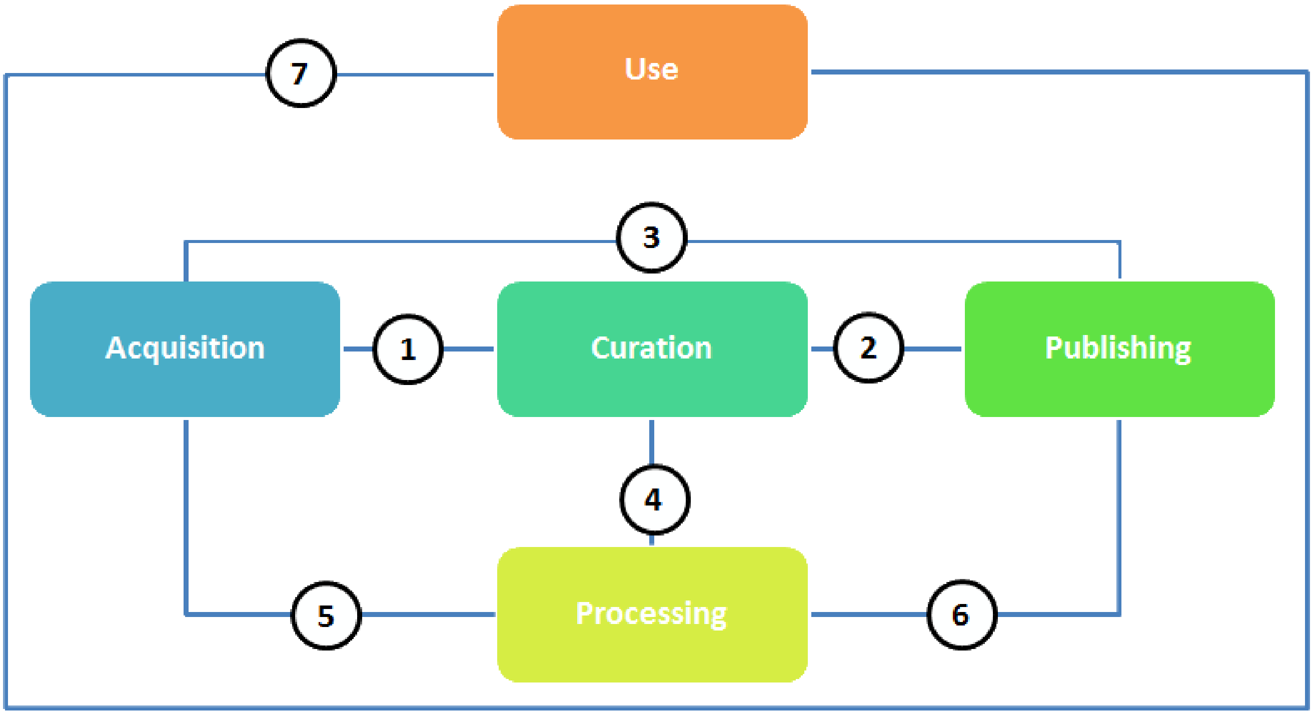


Illustration of the major integration (reference) points between different phases of the data lifecycle.

The integration points described as follows refer to the components supporting a phase of the data lifecycle. However, the components being integrated can be within the same research infrastructure or in different research infrastructures.

1. **Acquisition/Curation** by which components specialized in data acquisition are integrated with components which manage data curation.
2. **Curation/Publishing** by which components specialized in data curation are integrated with components which support data publishing.
3. **Acquisition/Publishing** by which components specialized in data acquisition are integrated components which support data publishing.
4. **Curation/Processing** by which components specialized in data curation are integrated with components which support data processing.
5. **Acquisition/Processing** by which components specialized in data acquisition are integrated with components which support data processing.
6. **Processing/Publishing** by which the components specialized in data processing are integrated with components which support data publishing.
7. **Use/All** by which entities outside the research infrastructure may be allowed to provide, access, or use data at different phases of the data lifecycle.

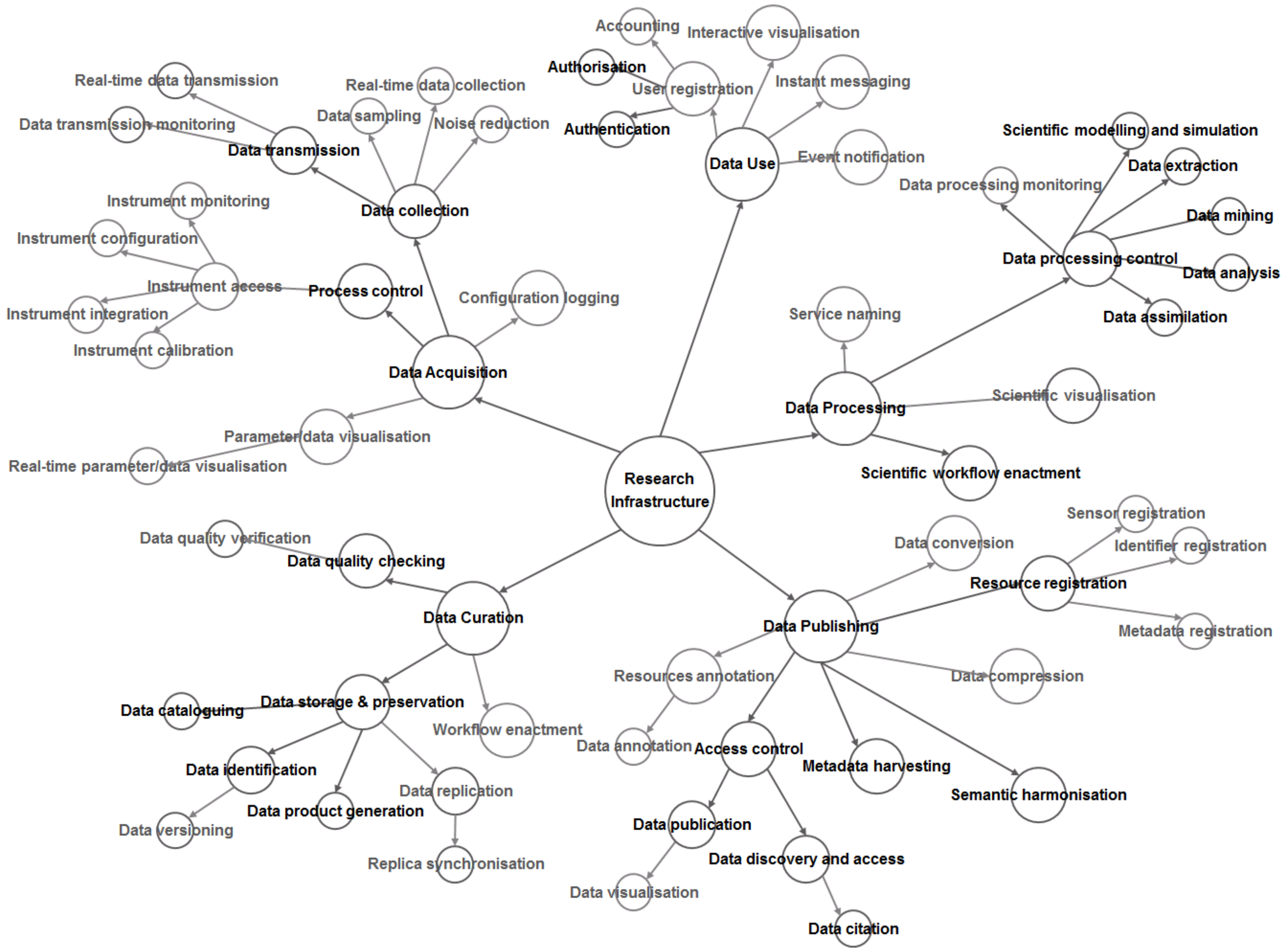
No notion of direction is implied in the definition of these points of reference. Relations with direction only appear when interfaces are superimposed on reference points, and then they can be unidirectional in either or both directions, or bidirectional - according to the nature of the interface(s).

Depending on the distribution of resources in an implemented infrastructure, some of these integration points may not be present in the infrastructure. They take particular importance however when considering scenarios where a research infrastructure delegates or outsources functionalities to other infrastructures. For example, EPOS and LifeWatch both delegate data acquisition and some data curation activities to national-level and/or domain-specific infrastructures, but provide data processing services over the data held by those infrastructures. Thus reference points 4 and 5 become of great importance to the construction of those projects.

Common Functions within a Common Lifecycle

Analysis of requirements of environmental research infrastructures during the ENVRI and ENVRIplus projects has resulted in the identification of a set of common functionalities. These functionalities can be classified according to the five phases of the data lifecycle. The requirements encompass a range of concerns, from the fundamental (*e.g.* data collection and storage, data discovery and access and data security) to more specific challenges (*e.g.,* data versioning, instrument monitoring and interactive visualisation).

In order to better manage the range of requirements, and in order to ensure rapid verification of compliance with the ENVRI-RM, a *minimal model* has been identified which describes the fundamental functionality necessary to describe an environmental research infrastructure. The minimal model is a practical tool to produce a partial specification of a research infrastructure which nonetheless reflects the final shape of the complete infrastructure without the need for significant refactoring. Further refinement of the models using the ENVRI-RM allow producing more refined models of designated priority areas, according to the purpose for which the models are created.



Radial depiction of ENVRI-RM requirements. The black labels correspond to the minimal model requirements.

The definitions of the minimal set of functions are given as follows (a full list of common functions is provided in Appendix A Common Requirements of Environmental Research Infrastructures):

(A) Data Acquisition

Process Control: Functionality that receives input status, applies a set of logic statements or control algorithms, and generates a set of analogue / digital outputs to change the logic states of devices.

Data Collection: Functionality that obtains digital values from a sensor instrument, associating consistent timestamps and necessary metadata.

Data Transmission: Functionality that transfers data over a communication channel using specified network protocols.

(B) Data Curation

Data Quality Checking: Functionality that detects and corrects (or removes) corrupt, inconsistent or inaccurate records from datasets.

Data Identification: Functionality that assigns (global) permanent unique identifiers to data products.

Data Cataloguing: Functionality that associates a data object with one or more metadata objects which contain data descriptions.

Data Product Generation: Functionality that processes data against requirement specifications and standardised formats and descriptions.

Data Storage & Preservation: Functionality that deposits (over the long-term) data and metadata or other supplementary data and methods according to specified policies, and then to make them accessible on request.

(C) Data Publishing

Access Control: Functionality that approves or disapproves of access requests based on specified access policies.

Metadata Harvesting: Functionality that (regularly) collects metadata in agreed formats from different sources.

Resource Registration: Functionality that creates an entry in a resource registry and inserts a resource object or a reference to a resource object with specified representation and semantics.

Data Publication: Functionality that provides clean, well-annotated, anonymity-preserving datasets in a suitable format, and by following specified data-publication and sharing policies to make the datasets publically accessible or to those who agree to certain conditions of use, and to individuals who meet certain professional criteria.

Data Citation: Functionality that assigns an accurate, consistent and standardised reference to a data object, which can be cited in scientific publications and/or from other data collections.

Semantic Harmonisation: Functionality that unifies similar data (knowledge) models based on the consensus of collaborative domain experts to achieve better data (knowledge) reuse and semantic interoperability.

Data Discovery and Access: Functionality that retrieves requested data from a data resource by using suitable search technology.

(D). Data Processing

Data Assimilation: Functionality that combines observational data with output from a numerical model to produce an optimal estimate of the evolving state of the system.

Data Analysis: Functionality that inspects, cleans, and transforms data, providing data models which highlight useful information, suggest conclusions, and support decision making.

Data Mining: Functionality that supports the discovery of patterns in large datasets.

Data Extraction: Functionality that retrieves data out of (unstructured) data sources, including web pages, emails, documents, PDFs, scanned text, mainframe reports, and spool files.

Scientific Modelling and Simulation: Functionality that supports the generation of abstract, conceptual, graphical or mathematical models, and to run an instances of those models.

Scientific Workflow Enactment: Functionality provided as a specialisation of Workflow Enactment supporting the composition and execution of computational or data manipulation steps in a scientific application. Important processing results should be recorded for provenance purposes.

Data Processing Control: Functionality that initiates calculations and manages the outputs to be returned to the client.

(E) Data use

Authentication: Functionality that verifies the credentials of a user.

Authorisation: Functionality that specifies access rights to resources.

The ENVRI Reference Model

The ENVRI Reference Model (ENVRI RM) defines an archetypical environmental research infrastructure. The ENVRI RM is structured according to the Open Distributed Processing (ODP) standard, ISO/IEC 10746-n, and as such, is defined from five different perspectives.

The Science, Information and Computational viewpoints take particular priority. These viewpoints allow expression of the complex concerns of the research infrastructures at a high level of abstraction. When building a research infrastructure, these viewpoints are important during the design and conceptualisation phases. These viewpoints have been defined previously by the ENVRI project, and enhanced by the ENVRIplus project.

The Engineering and Technology viewpoints complement the high level abstractions of the other three viewpoints by describing elements for physically building research infrastructures. When building a research infrastructure, these viewpoints are more relevant in the implementation and operational phases. These viewpoints are being defined as part of the ENVRIplus project.

- **Science Viewpoint**
- **Information Viewpoint**
- **Computational Viewpoint**
- **Engineering Viewpoint (Draft)**
- **Technology Viewpoint (Draft)**

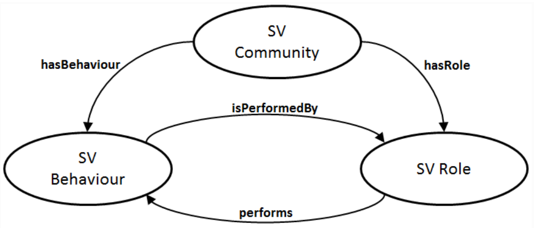
Science Viewpoint

The Science Viewpoint (SV) of the ENVRI RM captures the requirements for an environmental research infrastructure from the perspective of the people who perform their tasks and achieve their goals as mediated by the infrastructure. Modelling in this viewpoint derives the principles and properties of model objects through the analysis of the structure and functionality of organisations, people interacting within and around those organisations, and rules governing the interactions.

The Science Viewpoint defines communities with their community behaviours and roles. The diagram below shows the main elements of the science viewpoint and their relationships. Each ellipse contains a concept. The arrows connecting the concepts are directed and indicate the relationship between to concepts. The label of the link indicates the type of relationship. From this, the diagram indicates that SV behaviours are performed by SV roles. This is represented by two relationships, **isPerformedBy** and **performs**.

Two requirements engineering efforts in the ENVRI and ENVRIplus projects revealed the existence of a common lifecycle for the data produced, shared, and processed by research infrastructures. The five phases of the data lifecycle are *Data Acquisition*, *Data Curation*, *Data Publishing*, *Data Processing* and *Data Use*. Correspondingly, the **SV Community Behaviours** that support these five phases in order to collaboratively conduct scientific research, from data collection to the delivery of scientific results, can also be grouped in the same way. Such groups are called ***communities*** in ODP. The Science Viewpoint defines what those **SV Communities** are, what kind of **roles** they have, and what main **behaviours** they act out.

- **SV Communities**
- **SV Community Roles**
- **SV Community Behaviours**



Science Viewpoint objects and their relationships

SV Communities

A *community* is a collaboration which consists of a set of **roles** agreeing their **objective** to achieve a stated scientific purpose by means of a set of **SV Community Behaviours**.

The ENVRI RM distinguishes five groups of **SV Community Behaviours** and **roles** , seen as communities which by design align with the five phases of the **Model Overview**.

ODP defines **objective** as the abstraction of a system's purpose. An objective can be decomposed into **sub-objectives** which can be associated to different SV objects, for instance each community object can have a specific objective.

The ENVRI RM adheres to the following definition of Research Infrastructure:

> "Facilities, resources and related services that are used by the scientific community to conduct top-level research in their respective fields and covers major scientific equipment or sets of instruments; knowledge-based resources such as collections, archives or structures for scientific information; enabling Information and Communications Technology-based infrastructures such as Grid, computing, software and communication, or any other entity of a unique nature essential to achieve excellence in research. Such infrastructures may be “single-sited” or “distributed” (an organised network of resources)" [41]

In the ENVRI RM, a environmental research infrastructure is itself an **SV Object Actor**, a special type of **SV Object Classification.**

The purpose of the ENVRI RM is to provide a model of the IT infrastructure which is explicitly designed to fulfill the objectives of an ideal environmental research infrastructure.

The five communities are, *data acquisition*, *data curation*, *data publication*, *data service provision*, and *data use* The definitions of the communities are based on their **objectives**.

- **Data Acquisition Community**, collect raw data and bring (streams of) measurements into a system.
- **Data Curation Community**, curate the scientific data, maintain and archive them, and produce various data products with metadata.
- **Data Publishing Community**, assist data publication, discovery and access.
- **Data Processing Community**, provide various services, applications and software/tools to link and recombine data and metadata in order to derive knowledge.
- **Data Use Community**, make use of data and service products, and transfer knowledge into understanding.

In addition to the textual description, RI Communities can be in a diagram such as the one shown below. In the diagram the package represents an Environmental research infrastructure. The infrastructure consists of five communities and each community is linked to a community objective. The relationship between objective and sub-objectives is composition. Consequently, the objective of the RI is the composition of the objectives of its communities.

Note
The ENVRI RM defines five objectives which constitute the generic objective of an ideal RI. However, when modelling RIs may define additional objectives as needed.

SV Communities
Notation

The community roles and behaviours are described at the following pages:

- **SV Community Roles**
- **SV Community Behaviours**

SV Community Roles

A **role** in a community is a prescribing behaviour that can be performed any number of times concurrently or successively. A role can be either *active* (typically associated with a human actor) or *passive* (typically associated with a non-human actor, e.g. software or hardware components).

Active roles are identified in relation to people associated with a research infrastructure:

- those who use the research infrastructure to do science;
- those who work on resources to build, maintain and operate the research infrastructure; and
- those who govern, manage and administer the research infrastructure

Note
An individual may be a member of more than one community by undertaking different roles.

Passive roles are identified with subsystems, subsystem components, and hardware facilities. Active roles interact with passive roles to achieve their objectives.

- Research Infrastructure
- Roles in the Data Acquisition Community
- Roles in the Data Curation Community
- Roles in the Data Publishing Community
- Roles in the Data Processing Community
- Roles in the Data Use Community

Research Infrastructure

Research Infrastructure is the main entity being modeled in any ENVRI RM specification, additionally RI is a special role which can be part of the communities. This role is defined as follows.

Research Infrastructure: An active or passive role, which is the conglomeration of research resources providing some subset of data acquisition, data curation, data publishing, data processing and data use functionality to a research community.

Roles in the Data Acquisition Community

The main objective of the data acquisition community is to bring measurements into the system. Consider a typical data acquisition scenario: A measurement and monitoring model is designed by *designers* based on the requirements of *environmental scientists.* Such a design decides what data is to be collected and what metadata is to be associated with it, e.g., experimental information and instrument conditions. *Technicians* configure and calibrate a *sensor* or a *sensor network* to satisfy the experiment specifications. In the case where human sensors are to be used, *observers* or *measurers* input the measures to the system, e.g., by using mobile devices. *Data collectors* interact with a data acquisition subsystem to prepare the data or control the flow of data in order to automatically collect and transmit the data.

The following roles are identified in a data acquisition community:

- **Environmental Scientist**: An active role, which is a person who conducts research or performs scientific investigations. Using knowledge of various scientific disciplines, they may collect, process, analyse, synthesize, study, report, and/or recommend action based on data derived from measurements or observations of (for example) air, rock, soil, water, nature, and other sources.
- **Sensor**: A passive role, which is a converter that measures a physical quantity and converts it into a signal which can be read by an observer or by an (electronic) instrument.
- **Sensor network:** A passive role, which is a network consisting of distributed autonomous sensors to monitor physical or environmental conditions.
- **Measurement Model Designer**: An active role, which is a person who designs the measurements and monitoring models based on the requirements of environmental scientists.
- **Technician**: An active role, which is a person who develops and deploys sensor instruments, establishing and testing the sensor network, operating, maintaining, monitoring and repairing the observatory hardware.
- **Measurer**: An active role, which is a person who determines the ratio of a physical quantity (such as a length, time, temperature etc.), to a unit of measurement (such as the meter, second or degree Celsius).
- **Observer**: An active role, which is a person who receives knowledge of the outside world through his senses, or records data using scientific instruments.
- **Data collector**: Active or passive role, adopted by a person or an instrument collecting data.
- **Data Acquisition Subystem**: In the Science Viewpoint, the data acquisition subsystem is passive role of the data acquisition community. It is the part of the research infrastructure providing functionalities to automate the process of data acquisition.

Acquisition Community Roles

Notation

The behaviours of the data acquisition community is described at **Acquisition Behaviours**.

Roles in the Data Curation Community

The data curation community responds to provide quality data products and maintain the data resources. Consider a typical data curation scenario: when data is being imported into a curation subsystem, a *curator* will perform the quality checking of the scientific data. Unique identifiers will be assigned to the qualified data, which will then be properly catalogued by associating necessary metadata, and stored or archived. The main human roles interacting with or maintaining a data curation subsystem are *data curators* who manage the data and *storage administrators* who manage the storage facilities. Upon registering a digital object in a repository, its *persistent identifier (PID)* and the *repository* name or IP address is registered with a globally available system of *identification services (PID service)*. Users may subsequently present the *PID* to an *PID service* to learn the network names or addresses of repositories in which the corresponding digital object is stored. Here, we use a more general term "PID" instead of "*handle*", and identify the key roles involved in the data curation process.

We identified the following roles in this community:

- **Data Curator**: An active role, which is a person who verifies the quality of the data; annotates the data; catalogues, preserves and maintains the data as a resource; and prepares various required data products.
- **Semantic Curator**: An active role, which is a person who designs and maintains local and global conceptual models and uses those models to annotate the data and metadata.
- **Storage Administrator**: An active role, which is a person who has the responsibilities to design data storage, tune queries, perform backup and recovery operations, set up RAID mirrored arrays, and make sure drive space is available for the network.
- **PID Manager**: A passive role, a system or service that assigns persistent global unique identifiers to data and metadata products. The Manager invokes a external entity, the PID Service, to obtain the PIDs. The manager maintains a local catalogue of PIDs that are being used to reference data and metadata. If the data or metadata in the RI change location or are removed, the PID manager updates this information locally and informs the PID Service.
- **PID Service:** A passive role, a public system or service which can generate and assign persistent global unique identifiers (PIDs). The PID Service also maintains a public registry of PIDs for digital objects.
- **Storage System**: A passive role, which includes memory, components, devices and media that retain data and metadata for an interval of time.
- **Catalogue System**: A passive role, a catalogue system is a special type of storage system designed to support building logical structures for classifying data and metadata.

- **Data Curation Subsystem**: the data curation subsystem is a passive role of the data curation community. It is the part of the research infrastructure which stores, manages and ensures access to all persistent data and metadata produced within the infrastructure.

Note
PID Service was called PID Generator, howerver, the analysis of the Identification and Citation practices made evident that the generation can be done inside the RI (by PID Manager), shared between the RI and the PID Service, or completely delegated to a PID Service. Consequently the names were changed after version 2.1 of the ENVRI RM.

The PID generator does not disappear completely, it is a refinement (specialisation/subclass) which can be implemented by PID service or the PID Manager

Curation Community Roles

Notation

The behaviours of the data curation community are described at **Curation Behaviours.**

Roles in the Data Publishing Community

The objectives of the data publishing community are to publish data and assist discovery and access. We consider the scenarios described by Kahn's data publication model [34]: an originator, i.e., a user with digital material to be made available for public access, makes the material into a digital object. A digital object is a data structure whose principal components are digital material, or data, plus a unique identifier for this material (and, perhaps, other material). To get a unique identifier, the user requests one from an authorised PID service. A user may then deposit the digital object in one or more repositories, from which it may be made available to others (subject, to the particular item’s terms and conditions, etc.).

The published data are to be discovered and accessed by data consumers. A semantic mediator is used to facilitate the heterogeneous data discovery.

In summary, the following roles are involved in the data publication community:

- **Data Originator**: Either an active or a passive role, which provides the digital material to be made available for public access.
- **Data Repository**: A passive role, which is a facility for the deposition of published data.
- **Semantic Mediator**: A passive role, which is a system or middleware facilitating semantic mapping (i.e., executing mapping and translation rules), discovery and integration of heterogeneous data.
- **Data Publisher**: An active role, is a person in charge of supervising the data publishing processes.
- **Data Publishing Subsystem**: In the Science Viewpoint, the data publishing subsystem represents a passive role of the data publication community. It is the part of the research infrastructure enabling the discovery and retrieval of scientific data. The access to this subsystem could require authorisation at different levels for different roles.
- **Data Consumer**: Either an active or a passive role, which is an entity who receives and uses the data.
- **Metadata Harvester**: A pasive role, which is a system or service collecting metadata which supports the construction/selection of a global conceptual model and the production of mapping rules

Publishing Community Roles

Notation

The behaviours of the data publishing community are described at **Publishing Behaviours**.

Roles in the Data Processing Community

The data processing community provides various application services such as data analysis, mining, simulation and modelling, visualisation, and experimental software tools, in order to facilitate the use of the data. We consider scenarios of service oriented computing paradigm which is adopted by the ENVRI implementation model, and identify the key roles as below. These concepts are along the lines of the existing standards such as OASIS Reference Model for Service Oriented Architecture.

- **Data Provider**: Either an active or a passive role, which is an entity providing the data to be used.
- **Service**: A passive role, in which a functionality for processing data is made available for general use.
- **Service Consumer**: Either an active or a passive role, which is an entity using the services provided.
- **Service Provider**: Either an active or a passive role, which is an entity providing the services to be used.
- **Service Registry**: A passive role, which is an information system for registering services.
- **Capacity Manager**: An active role, which is a person who manages and ensures that the IT capacity meets current and future business requirements in a cost-effective manner.
- **Data Processing Subsystem**: In the Science Viewpoint, the data processing subsystem represents a passive role of the data processing community. It is the part of the research infrastructure providing services for data processing. These services could require authorisation at different levels for different roles.
- **Processing Environment Planner**: An active agent that plans how to optimally manage and execute a data processing activity using RI services and the underlying e-infrastructure resources (handling sub-activities such as data staging, data analysis/mining and result retrieval).

Processing Community Roles

Notation

The behaviours of the data processing community are described at **Processing Behaviours**.

Roles in the Data Use Community

The main role in the data use community is a *user* who is the ultimate consumer of data, applications and services. Depending on the purposes of use, a user can be one of the following active roles:

- **Scientist (synonym: Researcher)**: An active role, which is a person who makes use of the data and application services to conduct scientific research.
- **Engineer (synonym: Technologist)**: An active role, which is a person who develops and maintains the research infrastructure.
- **Educator (synonym: Trainer)**: An active role, which is a person who makes use of the data and application services for education and training purposes.
- **Policy Maker (synonym: Decision Maker)**: An active role, which is a person who makes decisions based on the data evidence.
- **Stakeholder (synonyms: Private Investor, Private Consultant)**: An active role, which is undertaken by a person who makes use of the data and application service for predicting markets so as to make business decisions on producing related commercial products.
- **Citizen (synonyms: General Public, Media)**: An active role, which is a person or organisation interested in understanding the knowledge delivered by an environmental science research infrastructure, or discovering and exploring the **knowledge base** enable d by the research infrastructure.
- **Citizen Scientist**: An active role, member of the general public who engages in scientific work, often in collaboration with or under the direction of professional scientists and scientific institutions (also known as amateur scientist).
- **Data Use Subsystem:** In the Science Viewpoint, the data use subsystem represents a passive role of the data use community. It is the part of the research infrastructure supporting the access of users to an infrastructure. The data use subsystem manages, and tracks user activities and supports users to conduct their roles in different communities.

Use Community Roles

Notation

The behaviours of the data use community are described at **Use Behaviours**.

SV Community Behaviours

A **behaviour** of a community is a composition of **processes** addressing specific scientific concerns. Behaviours are performed by **SV Community Roles**. In the ENVRI RM, the modelling of community behaviours is based on analysis of the common operations of research infrastructures which has resulted in **Appendix A Common Requirements of Environmental Research Infrastructures**. The community behaviours model focuses on **Model Overview#ref_minimalset**. A community behaviour can be either a single function or a composition of several functions from the function list.

A **process** is a collection of **steps** taking place in a prescribed manner and leading to an objective. A **step** is an abstraction of an action, used in a **process**, that leaves unspecified the objects that participate in that action. Steps are introduced in the definition because not all the action details need be specified in the composition.

> **Note**
> The Science viewpoint of the ENVRI RM only defines high level behaviours. A behaviour can be refined further by defining its constituting processes and steps, however, this task is left to the actual system designers.

- Behaviours of the Data Acquisition Community
- Behaviours of the Data Curation Community
- Behaviours of the Data Publishing Community
- Behaviours of the Data Processing Community
- Behaviours of the Data Use Community

Behaviours of the Data Acquisition Community

The key behaviours of the data acquisition community through the interaction of the community roles include:

- **Design Experiment:** A behaviour performed by a *Environmental Scientist* that designs the scientific experiment which motivates the data acquisition activities**.**
- **Design Measurement Model**: A behaviour performed by a *Measurement Model Designer* that designs the measurement or monitoring model based on scientific requirements.
- **Instrument Configuration**: A behaviour performed by a *Technician* that sets up a *sensor* or a *sensor network*.
- **Instrument Calibration**: A behaviour performed by a *Technician* that controls and records the process of aligning or testing a *sensor* against dependable standards or specified verification processes.
- **Data Collection**: A behaviour performed by a *Data Collector* that control and monitor the collection of the digital values from a *sensor* instrument or a human sensor, such as a *Measurer* or a *Observer*, associating consistent time-stamps and necessary metadata.

Acquisition Community Behaviours

Notation

The roles of the data acquisition community are described in **Acquisition Community Roles**.

Behaviours of the Data Curation Community

The main behaviours of the data curation community include:

- **Data Quality Checking**: A behaviour performed by a *Data Curator* that detects and corrects (or removes) corrupt, inconsistent or inaccurate records from data sets.
- **Data Preservation**: A behaviour performed by a *Data Curator* that deposits (over long-term) the data and metadata or other supplementary data and methods according to specified policies, and makes them accessible on request.
- **Data Product Generation**: A behaviour performed by a *Data Curator* that processes data against requirement specifications and standardised formats and descriptions.
- **Data Replication**: A behaviour performed by a *Storage Administrator* that creates, deletes and maintains the consistency of copies of a data set on multiple storage devices.
- **Data Identification**: A behaviour performed by a *PID manager* which provides a unique PID for data and metadata being curated.
- **Select or Build Local Conceptual Model**: A behaviour performed by a *Semantic Curator* which supports the annotation of data and metadata.
- **Data Annotation**: A behaviour performed by a *Semantic Curator* which supports the linking of data and metadata with a local conceptual model.

Curation Community Behaviours

Notation

The roles of the data curation community which are described at **Curation Roles**.

Behaviours of the Data Publishing Community

The data publishing community may perform the following behaviours:

- **Data Publication**: A behaviour that provides clean, well-annotated, anonymity-preserving datasets in a suitable format, and by following specified data-publication and sharing policies, to make the datasets accessible publicly or to those who agree to certain conditions of use, and to individuals who meet certain professional criteria.

- **Semantic Harmonisation**: A behaviour enabled by a *Semantic Mediator* that unifies similar data (knowledge) models based on the consensus of collaborative domain experts to achieve better data (knowledge) reuse and semantic interoperability.
- **Data Discovery and Access**: A behaviour enabled by a *Data Discovery and Access System* that retrieves requested data from a data resource by using suitable search technology.
- **Data Citation**: A behaviour performed by a *PID Manager* that assigns an accurate, consistent and standardised reference to a data object, in the same way as researchers routinely provide a bibliographic reference to printed resources. The RI publishing the data can define the citation contents such as authors, and dates for different citation styles.
- **Metadata Harvesting**: A behaviour performed by a metadata harvester to gather metadata from data objects in order to construct catalogues of the available information. A functionality that (regularly) collects metadata (in agreed formats) from different sources.

Publication Community Behaviours

Notation

The roles of the data publication community are described at **Publication Roles**.

Behaviours of the Data Processing Community

The following behaviours of the data processing community are modelled:

- **Coordinate Service**: A behaviour performed by a *Service Provider* to coordinate the actions of distributed applications in order to reach consistent agreement on the outcome of distributed transactions.
- **Compose Service**: A behaviour performed by a *Service Provider* to combine multiple services which can be achieved by either *Choreography* or *Orchestration*. **Service Choreography** is a collaboration between *Service Providers* and *Service Consumers*. **Service Orchestration** is the behaviour that a *Service Provider* performs internally to realise a service that it provides [Bibliography#ref35].
- **Customise Processing Environment**: A behaviour performed by a processing environment planner to enable a Data Processing Subsystem to prepare customised infrastructure and service platforms for managing specific data processing applications optimally, including the planning, provisioning and deployment sub-activities.
- **Describe Service**: A behaviour performed by a *Service Provider* to provide the information needed in order to use a service [**8**].
- **Register Service**: A behaviour performed by a *Service Provider* to make the service visible to *Service Consumers* by registering it in a service registry [**8**].

These are general behaviours of a service-oriented computing model. In the context of environmental science research infrastructures, a data processing community will focus on the implementation of domain special services, in particular those supporting **Data Assimilation, Data Analysis**, **Data Mining, Data Extraction, Scientific Modelling and Simulation, (Scientific) Workflow Enactment (**See **Terminology and Glossary** for the definitions of these functionalities**).**

Processing Community Behaviours

Notation

The roles of the data processing community are described at **Processing Roles**.

Behaviours of the Data Use Community

The data use community can be divided in two main groups: (1) the behaviours performed by active roles (human actors) and (2) the behaviours performed by passive roles (computer resources). The first group encompass the activities performed by human actors using the RI to interact with the different components of the RI. This can extend to all the actors in all the communities defined in the SV, in addition to the ones in the use community, these behaviours form a specialisation of the use community which can be called community support behaviours (or user support behaviours). The second group corresponds to the behaviours that enable the authorisation, authentication, and accounting of the activities of users, conforming a second specialisation which can be called AAAI behaviours.

- **Co-create**: A behaviour in which active roles from all communities participate in the design and planning of activities for the collection, preservation, analysis or publishing of research data.
- **Collaborate**: A behaviour performed by active roles which entails assisting/participating in some of the phases of the collection, preservation, analysis or publishing of research data.
- **Contribute**: A behaviour performed by active roles which entails directly collecting, preserving, analysing, or publishing research data held by the RI, according to a predefined protocol.

- **User Behaviour Tracking (sysnonym: real user monitoring)**: A behaviour enabled by a Data Use Subsystem that to track the Users. User Behaviour Tracking is the analysis of visitor behaviour on a website. The analysis of an individual visitor's behaviour may be used to provide options or content that relates to their implied preferences; either during a visit or in the future visits. Additionally, it can be user to track content use and performance.
- **User Profile Management**: A behaviour enabled by a *Data Use Subsystem* to support persistent and mobile profiles, where profiles will include preferred interaction settings, preferred computational resource settings, and so on.
- **User Working Space Management**: A behaviour enabled by a *Data Use Subsystem* to support work spaces that allow data, document and code continuity between connection sessions and accessible from multiple sites or mobile smart devices.
- **User Working Relationships Management**: A behaviour enabled by a *Data Use Subsystem* to support a record of working relationships, (virtual) group memberships and friends.
- **User Work Group Support**: A behaviour enabled by a *Data Use Subsystem* to support controlled sharing, collaborative work and publication of results, with persistent and externally citable PIDs.

Use Community Behaviours

Notation

The roles of the data use community are described at **Use Roles**.

SV Object Classification

Science Viewpoint Objects can be grouped according to the way they participate in a behaviour in three categories: **Actor**, **Artefact**, and **Resource** **Bibliography#Ref37**.

Actor: A science object that participates in performing a behaviour (Human, Software or Organisation).

Examples of Actor can be

- An specific Environmental Research Scientist, the real world person, e.g. Marine Biologist Tomas Moro.
- An specific **research institution**, real world organisation, such as The Marine Biological Association.
- An specific software system or service, real world functioning or planned system, for instance the ICOS carbon portal.

Artefact: A science object that is mentioned in a behaviour, but is not an active participant in it (participant in the sense of being used or performing a behaviour)

Examples of Resource can be:

- A data set in a particular state: raw, curated, annotated
- Metadata
- A physical sample: magnetic reading, film, photograph, or biological sample, which will be digitised by a RI for preservation and further processing (e.g. the biological samples from which a DNA sequence is obtained)

Resource: A science object that is essential for the performance of an action, requiring allocation or possibly being used up, respect to a behavior.

Examples of Resource can be:

- A pool of processing machines which is allocated to enable a behaviour
- A catalogue which is used as reference for a specific behaviour

It is important to remark that these qualifications are with respect to particular behaviours; an artefact in one behaviour may be an actor in another. For instance, if a sensor is allocated to a certain data collection activity during the "design experiment" behaviour, we can say that the sensor is a resource (allocated to be used). However, if that same sensor is modelled in the context of the "data collection" behaviour, the sensor is an actor (performing the action of collecting data).

Note
The ENVRI RM does not go into greater detail about the definition of instances of these objects because they will depend directly on the context in which the modelling is performed and the specific characteristics of the research infrastructure being modelled.

Information Viewpoint

The goal of the Information Viewpoint (IV) is to provide a common abstract model for the shared research data handled by the infrastructure. The focus lies on the data itself, without considering any platform-specific or implementation details. It is independent from the computational interfaces and functions that manipulate the data or the nature of technology used to store it. Similar to a high level ontology, the IV aims to provide a unique and consistent interpretation of the shared information objects of a particular domain.

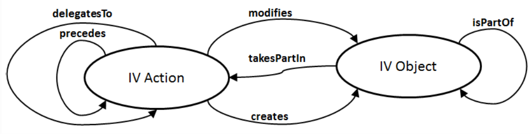
The IV specifies the types of the information objects and the relationships between those types. The main purpose of this viewpoint is to provide an abstract model of the lifecycles of the information objects handled by the RI. It also defines the constraints on information objects and the rules governing those lifecycles.

The models of the IV are grouped as follows.

- **IV Components**: collections of information objects and action types necessary to support the **Model Overview#ref_minimalset**.
- **IV Information Objects Lifecycle**: descriptions of how information objects change as the infrastructure operates,

The Information Viewpoint defines a set of IV objects and the set IV actions acting on those objects.

The diagram below shows the main elements of the IV and their relationships. Each ellipse contains a concept. The arrows connecting the concepts are directed and indicate the relationship between concepts. The label of the link indicates the type of relationship. From this, the diagram indicates that an IV object can be created by an IV action, as indicated by the **creates** relationship. Similarly, an IV object can be part of another IV object, as indicated by the **isPartOf** relationship. In this same way an action can be part of a chain of actions, this is indicated by two relationships **delegatesTo** and **precedes**.



Information Viewpoint components and their

illustrated using allowed state changes as the effects of the actions.
- **IV Information Management Constraints**: models of constraints that actions on information objects should implement to ensure the integrity and preservation of infor mation objects.

relationships

In the ENVRI RM research data and metadata are the main information objects managed by an RI. For this reason the IV is closely aligned with the **research data lifecycle model**.

IV Components

The ENVRI RM information viewpoint defines a configuration of information objects, the behaviour of those objects, the actions that operate on those objects, and a set of constraints that should always hold for actions applied on objects. The presentation of IV components are organised as follows:

- **IV Information Objects**: definition of a collection of information objects manipulated by the system.
- **IV Information Action Types**: definition of events that cause state changes of information objects.

The relationship between information objects and information actions is further refined in the following pages

- **IV Information Object Instances**: definition of how valid instances of information objects are modeled.
- **IV States**: description of information object states and state transitions resulting from actions.

IV Information Objects

The IV of the ENVRI RM defines two main types of information objects: Data and Metadata.

Information objects are used to model the various types of data and metadata manipulated by the RI. The IV information objects can be grouped as follows.

- Data: research data processed by the RI:
  - Persistent data data
  - Unique identifiers for the data identification
  - Backup (of data)
- Metadata: data typically related to the design of observation and measurement models, complements data by providing more precise details.
  - Design specification of the observation and measurement
  - Description of the measurement procedure
  - Quality Assurance (QA) annotations
  - Concepts from a conceptual model, e.g. an ontology
  - Mapping rules which are used for the model-to-model transformations
  - Provenance records
  - Management metadata(The data used to identify the states of data and metadata objects)

- Information Object Definitions
  - backup
  - mapping rule
  - citation
  - concept
  - conceptual model
  - data
  - data provenance
  - measurement result
  - metadata
  - metadata catalogue
  - metadata state
  - persistent data
  - persistent data state
  - qa notation
  - specification of investigation design
  - specification of measurements or observations
  - unique identifier (UID)

Information Object Types
Notation

Information Object Definitions

backup

A copy of (persistent) data so it may be used to restore the original after a data loss event.
mapping rule

Configuration directives used for model-to-model transformation.

Mapping rules can be transformation rules for:

- arithmetic values (mapping from one unit to another)
  from linear functions like k.x + d to multivariate functions
- ordinal and nominal values
  e.g. transforming classifications according to a classification system A to classification system B
- data descriptions (metadata or **Semantic Annotation** or QA annotation)
- parameter names and descriptions (can be n:m)
- method names and descriptions
- sampling descriptions

citation

A published, resolvable, token linking to a persistent data object via an identifier.

In information technology terms, a citation is a reference to published data which may include the information related to:

- the data source(s)
- the owner(s) of the data source(s)
- a description of the evaluation process, if available
- a timestamp marking the access time to the data sources, thus reflecting a certain version
- the equipment used for collecting the data (individual sensor or sensor network)

It is important that the citation is resolvable, which means that the identifiers point to live data sets and that the meaning of the items above are made clear.

concept

Identifier, name and definition of the meaning of a thing (abstract or real thing). Human readable definition by sentences, machine readable definition by relations to other concepts (machine readable sentences). It can also be meant for the smallest entity of a conceptual model. It can be part of a flat list of concepts, a hierarchical list of concepts, a hierarchical thesaurus or an ontology.

conceptual model

A collection of concepts, their attributes and their relations. It can be unstructured or structured (e.g. glossary, thesaurus, ontology). Usually the description of a concept and/or a relation defines the concept in a human readable form. Conceptual models can also be represented in machine readable formats, for instance RDFS or OWL. Those sentences can be used to construct a self description. It is common practice to provide both the human readable description and the machine readable description within the same system. In this sense, a conceptual model can also be seen as a collection of human and machine readable sentences. They can be local, developed within a project, or global, accepted and used by a wider community (such as GEMET or OBOE). Conceptual models can be used to annotate data (e.g. within a network of triple stores).

data

Research data processed by the RI. This is the base information object class from which all other information objects are derived

data provenance

Metadata that traces the origins of data and records all state changes of data during their lifecycle and their movements between storages.

A creation of an entry into the data provenance records triggered by any actions typically contains:

- date/time of action;
- actor;
- type of action;
- data identification.

Data provenance system is an annotation system for managing data provenances. Usually unique identifiers are used to refer the data in their different states and for the description of the different states.

measurement result

Quantitative, qualitative, or cataloguing determinations of magnitude, dimension, and uncertainty to the outputs of observation instruments, sensors, sensor networks, human observers and observer networks.

metadata

Data about data, in scientific applications is used to describe, explain, locate, or make it easier to retrieve, use, or manage a data resource.

There have been numerous attempts to classify the various types of metadata. As one example, NISO (National Information Standards Organisation) distinguishes between three types of metadata based on their functionality: Descriptive metadata, which describes a resource for purposes, such as discovery and identification; Structural metadata, which indicates how compound objects are put together; and Administrative metadata, which provides information to help manage a resource. But this is not restrictive. Different applications may have different ways to classify their own metadata.

Metadata is generally encoded in a metadata schema which defines a set of metadata elements and the rules governing the use of metadata elements to describe a resource. The characteristics of metadata schema normally include: the number of elements, the name of each element, and the meaning of each element. The definition or meaning of the elements is the semantics of the schema, typically the descriptions of the location, physical attributes, type (i.e., text or image, map or model), and form (i.e., print copy, electronic file). The value of each metadata element is the content. Sometimes there are content rules and syntax rules. The content rules specify how content should be formulated, representation constraints for content, allowable content values and so on. And the syntax rules specify how the elements and their content should be encoded. Some popular syntax used in scientific applications include Some popular syntax includes:

- HTML (Hyper-Text Markup Language): **www.w3.org/MarkUp/**
- XML (eXtensible Markup Language): **www.w3.org/XML/**
- RDF (Resource Description Framework): **www.w3.org/RDF/**
- OWL (Web Ontology Language): **www.w3.org/2001/sw/**
- SGML (Standard Generalised Markup Language): **www.w3.org/MarkUp/SGML/**
- MARC (Machine Readable Cataloging): **www.loc.gov/marc/**
- MIME (Multipurpose Internet Mail Extensions): **www.ukoln.ac.uk/metadata/resources/mime/**

- DIME (Direct Internet Message Encapsulation): **xml.coverpages.org/draft-nielsen-dime-01.txt**

Such syntax encoding allows the metadata to be processed by a computer program.

Many standards for representing scientific metadata have been developed within disciplines, sub-disciplines or individual project or experiments. Some widely used scientific metadata standards include:

- Dublin Core: **purl.oclc.org/metadata/dublin_core/**
- CERIF (Common European Research Information Format): **www.eurocris.org**
- ISO 11179: **metadata-stds.org/11179/**
  ISO 19115 (by iso-tc 211): **www.isotc211.org**
- FGDC (The Federal Geographic Data Committee): **www.fgdc.gov/standards**
- INSPIRE: **inspire.jrc.ec.europa.eu**
- ISO 19115, Geographic information - metadata standard (metadata model closely related to INSPIRE) **www.iso.org**
- DDI (Data Documentation Initiative): **www.ddialliance.org**
- TEI (The Text Encoding Initiative): **www.tei-c.org**
- METS (Metadata Encoding and Transmission Standard): **www.loc.gov/standards/mets**
- MODS (Metadata Object Description Schema): **www.loc.gov/standards/mods/**
- OAIS (Reference Model for an Open Archival Information System)

Two aspects of metadata give rise to the complexity in management:

- Metadata are data, and data become metadata when they are used to describe other data. The transition happens under particular circumstances, for particular purposes, and with certain perspectives, as no data are always metadata. The set of circumstances, purposes, or perspectives for which some data are used as metadata is called the ‘context’. So metadata are data about data in some ‘context’.
- Metadata can be layered. This happens because data objects may move to different stages during their life in a digital environment requiring their association to different layers of metadata at each stage.

Metadata can be fused with the data. However, in many applications, such as a provenance system or a distributed satellite image annotation system, the metadata and data are often created and stored separately, as they may be generated by different users, in different computing processes, stored at different locations and in different types of storage. Often, there is more than one set of metadata related to a single data resource, e.g. when the existing metadata becomes insufficient, users may design new templates to make another metadata collection. Efficient software and tools are required to facilitate the management of the linkage between metadata and data. Such linkage relationship between metadata and data are vulnerable to failures in the processes that create and maintain them, and to failures in the systems that store their representations. It is important to devise methods that reduce these failures.

metadata catalogue

A collection of metadata, usually established to make the metadata available to a community. A metadata catalogue can be exposed through an access service.

metadata state

metadata state is an object property that determines the set of all sequences of actions (or traces) in which the metadata object can participate, at a given instant in time (as defined in ODP, ISO/IEC 10746-2).

In their lifecycle, metadata may have the states described in the following table.

State	Description
raw	metadata which are not yet registered or organised in a catalogue. Raw metadata are not shareable in this status.
registered	metadata which have been stored in a metadata catalogue.
annotated	metadata that are associated to concepts, describing their meaning
published	metadata made available to the public, the outside world. Metadata registered within public catalogues.

persistent data

Data is the representations of information dealt with by information systems and users thereof (as defined in ODP, ISO/IEC 10746-2). Persistent Data denotes data that are persisted (stored for the long-term).

persistent data state

Persistent Data state is an object property that determines the set of all sequences of actions (or traces) in which the object can participate, at a given instant in time (as defined in ODP, ISO/IEC 10746-2). The persistent data states and their changes as effects of actions are illustrated as **IV States**.

In their lifecycle, persistent data may have the states described in the following table.

State	Description
raw	data derived from the primary results of observations or measurements
identified	data which has been assigned a unique identifier
annotated	data that are associated to concepts, describing their meaning

qa assessed	data that have undergone checks and are associated with descriptions of the results of those checks.
assigned metadata	data that are associated to metadata which describe those data
backed up	data that of which an identical copy has been stored securely
finally reviewed	data that have undergone a final review and therefore will not be changed any more
mapped	data that are mapped to a certain conceptual model
published	data that are presented to the outside world
processed	data that have undergone a processing (evaluation, transformation)

> **Note**
> The state 'raw' refers to data as received into the ICT elements of the research infrastructure. Some pre-processing may or may not have been carried out closer to where measurements and observations were made

These states are referential states. The instantiated chain of data lifecyle can be expressed in data provenance.

qa notation

Notation of the result of a Quality Assessment. This notation can be a nominal value out of a classification system up to a comprehensive (machine readable) description of the whole QA process.

In practice, this can be:

- simple flags like "valid" / "invalid" up to comprehensive descriptions like
- "data set to invalid by xxxxxx on ddmmyy because of yyyyyyy"

QA notation can be seen as a special annotation. To allow sharing with other users, the QA notation should be unambiguously described so as to be understood by others or interpretable by software tools.

service description

Description of services and processes available for reuse. The description is needed to facilitate usage. The service description usually includes a reference to a service or process making it available for reuse within a research infrastructure or within an open network like the Internet. Usually such descriptions include the accessibility of the service, the description of the interfaces, the description of behavior and/or implemented algorithms. Such descriptions are usually done along service description standards (e.g. WSDL, web service description language). Within some service description languages, semantic descriptions of the services and/or interfaces are possible (e.g. SAWSDL, Semantic Annotations for WSDL).

specification of investigation design

This is the background data needed to understand the overall goal of the measurement or observation. It could be the sampling design of observation stations, the network design, the description of the setup parameters (interval of measurements) and so on. It usually contains important data for the allowed evaluations of research results (e.g. the question of whether a sampling design was done randomly or by stratification determines which statistical methods can be applied).

Investigations (and hence measurement and observation results) need not be quantitative. They can also be qualitative results (like "healthy", "ill") or classifications (like assignments to biological taxa). It is important for data processing to know whether they are quantitative or qualitative.

The specification of investigation design can be seen as part of metadata or as part of the **Semantic Annotation**. It is important that this description follows certain standards and it is desirable that the description is machine readable.

specification of measurements or observations

The description of the measurement/observation which specifies:

- what is measured/observed;
- how it is measured/observed (including processes/metods and instruments to be used);
- by whom it is measured/observed (including project, organisation and experimenter/observer profile); and
- what the temporal design is (single / multiple measurements / interval of measurement etc.)

> **Note**
> This specification can be included as metadata or as **Semantic Annotation** of the scientific data to be collected. It is important that such a design specification is both explicit and correct, so as to be understood or interpreted by external users or software tools. Ideally, a machine readable specification is desired.

unique identifier (UID)

With reference to a given type of data, objects a unique identifier (UID) is any identifier which is guaranteed to be unique among all identifiers used for those type of objects and for a specific purpose.

There are 3 main generation strategies:

- serial numbers, assigned incrementally;

- random numbers, selected from a number space much larger than the maximum (or expected) number of objects to be identified. Although not really unique, some identifiers of this type may be appropriate for identifying objects in many practical applications and are, with abuse of language, still referred to as "unique";
- names or codes allocated by choice which are forced to be unique by keeping a central registry.

The above methods can be combined, hierarchically or singly, to create other generation schemes which guarantee uniqueness.

In many cases, a single object may have more than one unique identifier, each of which identifies it for a different purpose. For example, a single object can be assigned with the following identifiers:

- global: unique for a higher level community
- local: unique for the subcommunity

The critical issues of unique identifiers include but not limited to:

- long term persistence – without efficient management tools, UIDs can be lost;
- resolvability -- without efficient management tools, the linkage between a UID and its associated contents can be lost.

IV Information Action Types

IV actions model the processing information objects in the system. Every action is associated with at least one object. Actions cause state changes in the objects that participate in them.

The figure shows a collection of action types specified in the information viewpoint.

Information Action Types

Notation

IV Actions

Information Action Definitions

add metadata

Add additional data according to a predefined schema (metadata schema). This partially overlaps with data annotations.

annotate

Perform **Annotation** of an information object. Adding structured or unstructured information to describe a data object.

There are two basic types of annotation: **free text annotation** and s**emantic annotation**. free text annotation refers to adding short explanations or opinions to a text or drawing (dictionary definition). Semantic annotation refers to linking data to structured conceptual model

(ontology)

annotate data

Semantic annotation of persistent data with concepts of predefined local or global conceptual models.

In practices, this can be done by adding tags or a pointer to concepts within a conceptual model to the data. If the concepts are terms e.g., in an SKOS/RDF thesaurus, and published as linked data, then data annotation would mean to enter the URL of the term describing the meaning of the data.

There is no exact borderline between metadata and semantic annotation.

annotate metadata

Semantic annotation of metadata with concepts of predefined local or global conceptual models.

This can be done by adding pointers from concepts within a conceptual model to the metadata. For instance, if concepts are terms of a SKOS thesaurus, identified by URIs and published as linked data, then annotation amounts to associating metadata with the terms' URIs.

assign unique identifier

Obtain a unique identifier and associate it to the data.

build conceptual models

Establish a local or global model of interrelated concepts.

This may involve the following issues:

- commitment: the agreement of a larger group of scientists / data providers / data users should be achieved;
- unambiguousness: the conceptual model should be unambiguously defined;
- readability: the model should be readable by both human and machine. Ontologies, for instance, express the meaning of the concepts with the relations to other concepts while being human and machine readable. Recently it has increasingly become important to add definitions in human readable language.
- availability: the conceptual model must be referenceable and dereferenceable for a long time

carry out backup

Replicate data to an additional data storage so it may be used to restore the original after a data loss event. Long-term preservation is a special type of backup.

check quality

Actions to verify the quality of data.

For example it may involve:

- remove noise
- remove apparently wrong data
- calculate calibrations

Quality checks can be carried out at different points in the chain of data lifecycle.

Quality checks can be supported by software tools for those processes which can be automated (e.g. statistic tolerance checks).

describe service

Describe the accessibility of a service or processes, which is available for reuse, the interfaces, the description of behavior and/or implemented algorithms

do data mining

Execute a sequence of metadata / data request --> interpret result --> do a new request

Usually this sequence helps to deepen the knowledge about the data. Classically this sequence can:

- lead from data to metadata and semantic annotations
- follow the provenance of data
- can follow data processing

It can be supported by special software that helps to carry out that sequence of data request and interpretation of results.

final review

Review the data to be published, which will not likely be changed again.

The action triggers the change of the data state to be "finally reviewed". In practices, an annotation for such a state change should be recorded for provenance purposes. Usually, this is coupled with archiving and versioning actions.

free text annotation

To add a short explanation or opinion to an information object.

perform mapping

Execute transformation rules for values (mapping from one unit to another unit) or translation rules for concepts (translating the meaning from one conceptual model to another conceptual model, e.g. translating code lists).

perform measurement or observation

Measure parameter(s) or observe an event. The performance of a measurement or observation produces measurement results.

process data

Process data for the purposes of:

- converting and generating data products
- calculations: e.g., statistical processes, simulation models
- visualisation: e.g., alpha-numerically, graphically, geographically

Data processes should be recorded as provenance instances.

publish data

Make data public accessible.

For example, this can be done by:

- presenting them in browsable form on the world wide web
- by presenting them via special services:
- RESTful service
- SOAP service
- OPEN GRID service
- OGC service (web feature service, web map service)
- SPARQL endpoint

publish metadata

Make the registered metadata available to the public.

query data

Send a request to a data store to retrieve required data.

In practice, there are two types of data query:

- two step approach:

  step 1: query/search metadata;

  step 2: access data

  For example, when using OGC services, it usually first invokes a web feature service to obtain feature descriptions, then a web map service can be invoked to obtain map images.

- one step approach: to query data e.g., by using SQL services or SPARQL endpoints

  Requests can be directly sent to a service or distributed by a broker.

query metadata

Send a request to metadata resources to retrieve metadata of interests.

register metadata

Enter the metadata into a metadata catalogue.

resolve annotation

Retrieve the reference to the specific set of objects that correspond to a set of annotation terms.

semantic annotation

Linking of data objects whit structured information from predefined local or global conceptual models.

setup mapping rules

Specify the mapping rules of data and/or concepts.

These rules should be explicitly expressed using a language that can be processed by software.

A minimal set of mapping rules should include the following data:

- source data / concept for which the mapping is valid
- target data / concept for which the mapping is valid
- mapping process (the translation and/or transformation process)
- validity constraints for the mapping (temporal constraints, context constraints, etc.)

specify investigation design

specify design of investigation, including sampling design:

- geographical position of measurement or observation (site) -- the selections of observations and measurement sites, e.g., can be statistical or stratified by domain knowledge;
- characteristics of site;
- preconditions of measurements.

specify measurement or observation

Specify the details of the method of observations/measurements.

For example, it may include the specification of a measurement device type and its settings, measurement/observation intervals.

store data

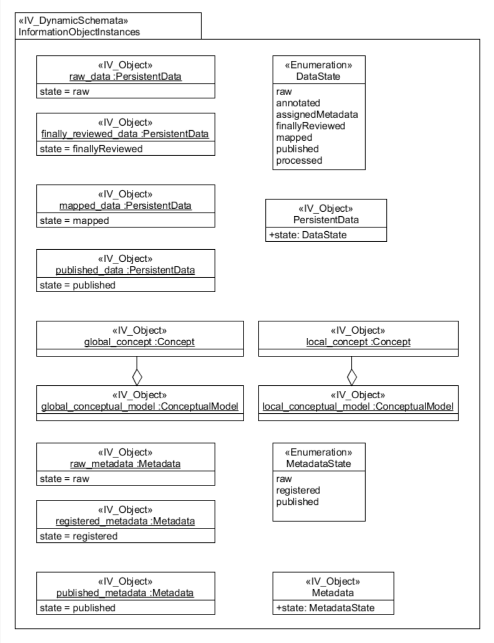
Archive or preserve data in persistent manner to ensure continued accessibility and usability.

track provenance

Automatically generate and store metadata about the actions and the data state changes as provenance instances.

IV Information Object Instances

Information Object Instances



Notation

Information object instances are used to define valid instances of information objects. As explained earlier an information object can have several state transitions. An information object instance is a model of an information object at a particular state.

The diagram on the right shows examples of different object instances. The main difference with information object is that the status of the information object instances is assigned a value from the list of allowed states.

Information objects instances are needed for two purposes:

1. to show the data state changes as effects of actions;
2. to show the relations between valid states of related data types, for instance that reaching the "published" requires a specific series of previous states which can be traced for QA and provenance validation.

The diagram also includes two types of conceptual models: "local conceptual model" and "global conceptual model".

Global conceptual model

A set of concepts accepted by a data sharing community.

Global conceptual models contain global concepts. Examples of these types of models are global Thesauri like GEMET / EuroVoc / AGROVOC or global ontologies like Gene Ontology.

Local conceptual model

A set of concepts locally agreed by a limited user community, such as the members of a research institution.

A local conceptual model contains concepts which have a specific meaning according to the community using them for instance local definitions of person, institute, or data.

A conceptual model can be local or global depending on the size of the community which commits to it. Local conceptual models can contain concepts borrowed from global conceptual models. Alternatively mapping rules can be established to determine equivalences between global and local concepts.

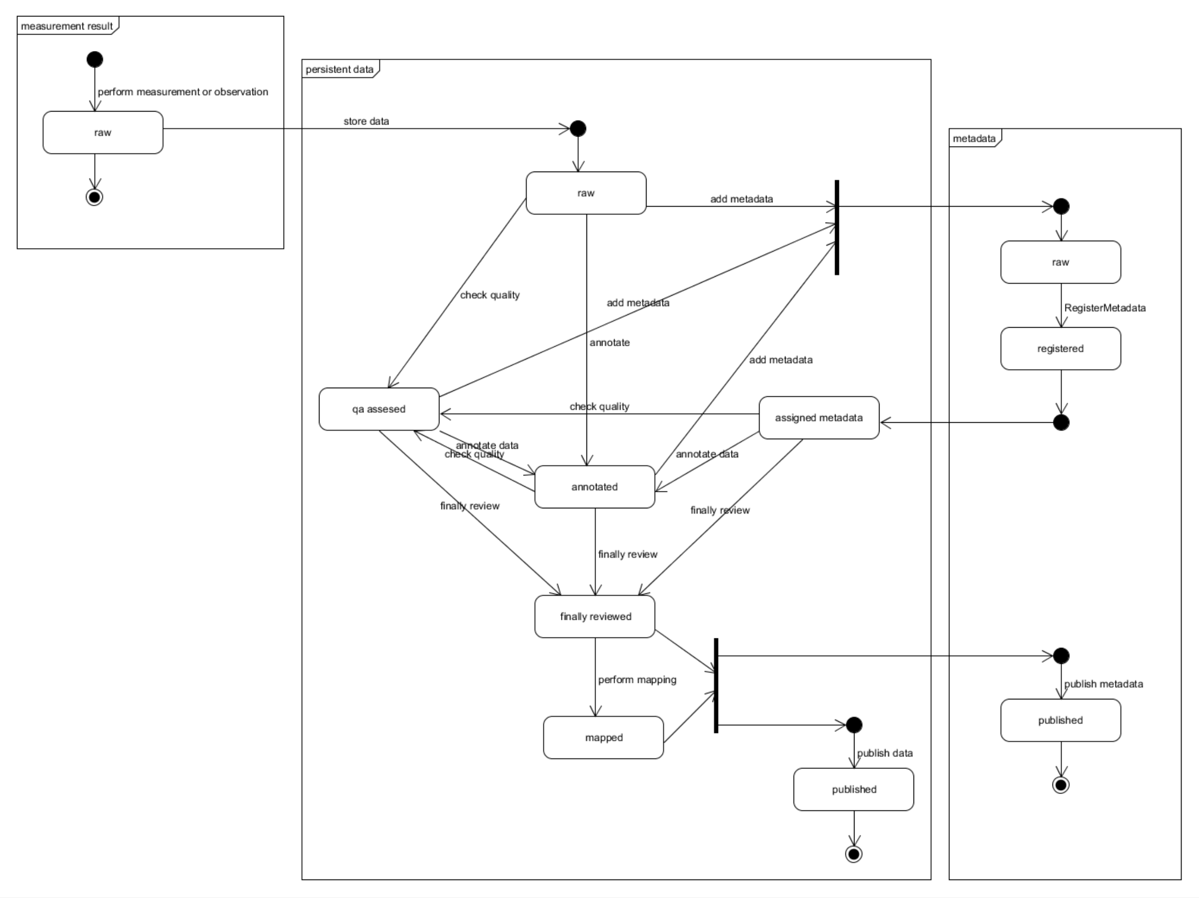
IV States

The ENVRI RM IV defines **data state** and **metadata state** as the set of attributes which determine the actions that can be performed over a given information object.

The state changes, together with the **IV Information Action Types** can be used to model the behaviour of data as it is managed by the RI.

The diagram below shows the states of three IV objects (Measurement Result, Persistent Data and Metadata) and their relationships.

State Diagrams



Notation

This diagram shows all the possible states for each of the IV Objects.

The first IV object (left) is Measurement Result object. The object is created from a **PerformMeasurementOrObservation** action. The Measurement Result has only one state "raw".

The second IV object (middle) is PersistentData object. The **StoreData** action triggers the creation of the new Persistent Data object in the RI which has a raw state. This object can have six states. The transitions shown in the figure indicate possible transitions which can occur during the lifetime of the object.

The third IV object (right) is a "Metadata" object. The **AddMetadata** action triggers the creation of the new Metadata object in the RI. This object can have three states. However, the transitions from registered to published is not directly triggered. The diagram indicates that publish metadata must occur simultaneously with publish data

In the diagram the filled circle indicates a starting point or a junction. Used as starting point, a filled circle indicates a pseudo-state which represents the start of the lifetime of an object instance. Used as a junction, the filled circle indicates a pseudo-state where paths merge or split. The Rectangles with a label in the upper left corner are used to indicate the object whose states are represented. The rectangles with rounded corners are used to indicate states. Each arrow indicates a transition between states. The label of the arrow indicates the activity that triggers the transition. The bar figure in the diagram indicates a pseudo-state that can represent a fork or a merge of paths. The type of diagram presented is an UML state machine diagram [**40**].

Simple use example

The diagram shows the series of actions applied change the state of the IV object until it reaches a "published" state. In this diagram, only the persistent data object is shown. The diagram is linear and depicts contains only a subset of the allowed states and actions. This subset can vary from one RI to another. RIs can model their data lifecycles using state machine diagrams. The states described in the IV States diagram indicate a set of possible paths to follow when representing the data lifecycle. The instances developed by RIs may chose the states they need to include to represent their corresponding data lifecycles. RIs can also include additional states which help them better represent their data processing flows.

State Diagram - Simple use example

Notation

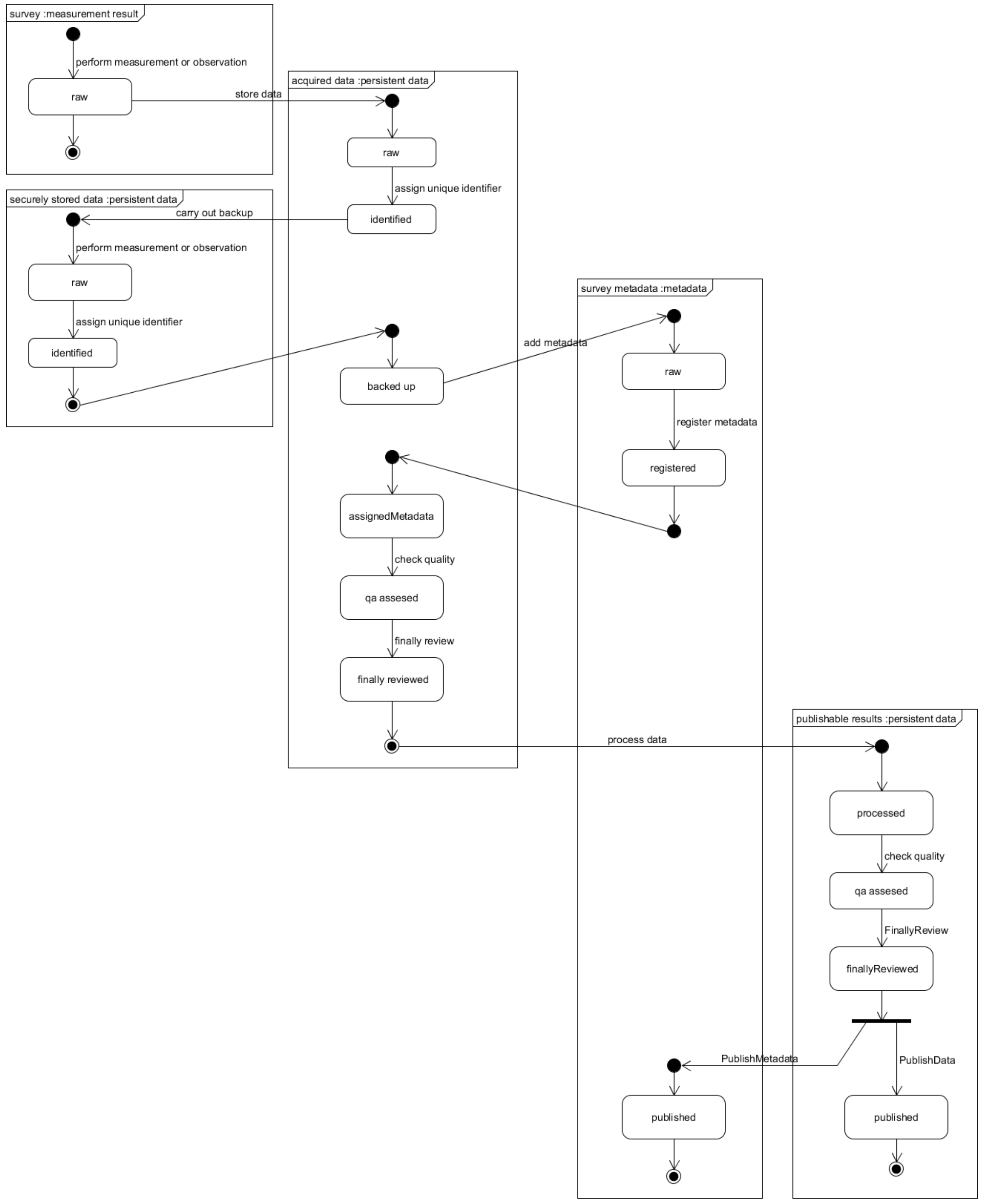
Advanced use example

The diagram shows the lifecycles of five IV objects and the way in which they relate to each other. The example is more complex but it is still linear and easy to follow.

Reading the diagram from the left to right and top to bottom it is possible to describe the lifecycles of the five IV objects. The first object is a called Survey. The diagram indicates that Survey is a type of MeasurementResult, when the Survey object is stored a new object is created SurveyData, a type of PersistentData object. The SurveyData object is then Identified, after identifying, a back up copy of the object, named SecuredStoredData, is created. Once the SecuredStoredData is stored and identified, the state of the SurveyData object changes to backedup. After backup, the RI adds metadata to the SecuredData object. The SurveyMetadata object is created and registered. This causes the change in the state of the SurveyData object to assignedMetadata. Subsequently the object is QA assesed and reviewed. Once the SurveyData is finally reviewed, the RI creates a new PersistentData object, PublishableResult, which results from the processing of the SurveyData object. The published result data object is then QA assesed and finally reviewed. The final pair of activities publish both the PublishableResult and the SurveyMetadata objects.

In this example, most of the states and actions are still subsets from the ones originally introduced on he IV State Diagram. The only exception is the backedup state. This is still valid, RIs can adapt the diagrams to their particular needs adding states and actions to better illustrate their data lifecycles as close to reality as possible.

State Diagram - Advanced use example



Notation

IV Information Objects Lifecycle

The specification of the lifecycles of the information objects is described combining IV object instances at different states and the sequences of allowed actions according to those states. The set of models used for describing this evolution are part of the dynamic schemata in ODP [37].

The specification of information objects lifecycle is presented in two parts:

- **IV Lifecycle Overview**: overview of information objects state changes as effects of actions.

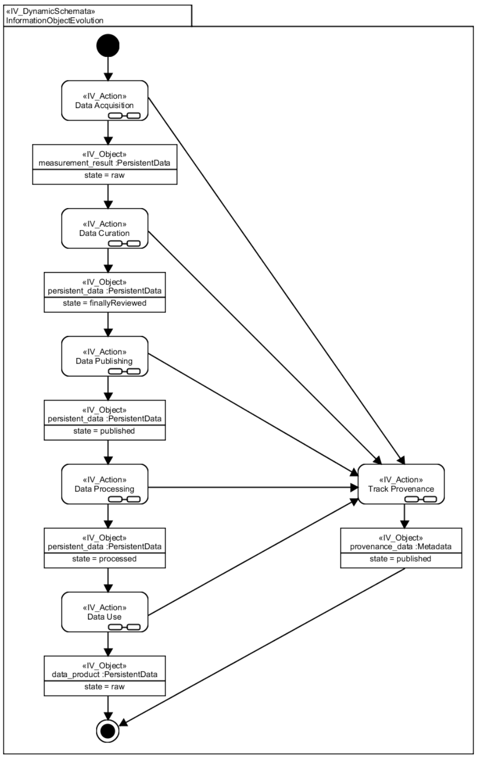
- **Lifecycle in detail** : detailed description of how information objects changes at each phase of the data lifecycle.

IV Lifecycle Overview

This section describes the alignment between data processing in the RI systems and the data lifecycle using **IV Information Objects** and **IV Information Action Types**. The description is framed against the phases of the **Model Overview**.

The diagram shown on the right provides a high level view of the data lifecycle. The rounded rectangles represent IV actions on data and the straight rectangles represent instances of IV objects at different states. The arrow lines link IV actions and IV objects as follows: arrows leaving an action connect to IV objects created by the action while arrows entering an action connect IV objects to actions applied on them. The black circle at the top of the diagram represents the starting point and the double circle at the bottom represents the end point. The types of diagrams used in this section are called activity diagrams (UML).

Information Object Lifecycle



Notation

> In the diagram each phase of the data lifecycle is represented as an action which produces a specific information object, in this case the main information object shown is persistent data. The diagram also adds a provenance tracking action. Provenance tracking is an action that can proceed in parallel during all phases of the data lifecycle. The overview of the data lifecycle phases is described as follows.

Data Acquisition: The data acquisition phase encompasses the actions defined for the observation/experimentation, storage, identification and storage of measurements/observations (raw data). In the diagram, the acquisition phase is represented by the "DataAcquisition" action which produces a measurement result data object with the state raw.

Data Curation: The data curation phase encompasses the actions that support the long term preservation and use of research data. The main product of this set of actions is persistent data in a stable state (curated data). In the diagram, the curation phase is represented by the "DataCuration" action which produces a persistent data object with the state curated.

> **Note**
> Data curation includes preservation which may require data transformation, for example media migration to a digital form.

Data Publishing: The data publishing phase encompasses the actions that guaranty data access and discovery for entities (people and systems) outside the RI. In the diagram, the publishing phase is represented by the "DataPublishing" action which produces a persistent data object with the state published.

Data Processing: The data processing phase encompasses the actions that support making use of the RI published data. In the diagram, the processing phase is represented by the "DataProcessing" action which produces a persistent data object with the state processed.

Data Use: The data use phase is a bridge phase which sits between processing and acquisition. In this phase, the data is used and may produce new data (raw data) which can in turn be persisted by an RI. In the diagram the usage phase is represented by the "DataUse" action which produces a data product object with the state raw.

In the **IV Lifecycle in Detail** section, the actions in the diagram are expanded to present a more detailed view of the data lifecycle from the IV perspective.

Data Provenance Tracking

It is important to track state changes of information objects during their lifecyle. As illustrated in diagram above, the ProvenanceTracking action takes place in parallel to the phases of the lifecycle that change the state of persistent data.

Some of the states changes of information objects as effects of actions are summarised in the following table. As shown in the diagram, the outputs of each transition in which a new stable state is reached can be used to produce provenance data. For example, a provenance tracking service may record information objects being processed, action types applied and resulting objects, the timestamps for the actions, and some additional data and store that as provenance data.

Simplified example of some provenance tracking points

Information Object	Applied Action Types	Resulting Information Objects
	Data Acquisition	persistent data (raw)
persistent data (raw)	Data Curation	persistent data (finallyReviewed) metadata (registered)
persistent data (FinallyReviewed) metadata(registered)	Data Publishing	persistent data (published) metadata (published)
persistent data (published)	Data Processing	persistent data (processed)
persistent data (processed)	Data Use	data product (new form of persistent data (raw))

The citation of data referencing the actors of involved in production of the data is an example of the use of data provenance

Correct interpretation of the data can also depend on reviewing the provenance, for instance to ensure origin of the data matches its intended use.

IV Lifecycle in Detail

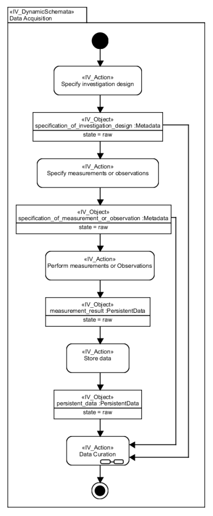
This section expands the **IV Lifecycle Overview** of the alignment between the information viewpoint and the data lifecycle. The descriptions uses the **IV Information Objects** and **IV Information Action Types** to a greater extent providing a deeper insight into the processing of information objects by the RI.

The notation for the diagrams in this section is as follows. The rounded rectangles represent IV actions on data and the straight rectangles represent instances of information objects at different stages. The arrow lines link actions and objects as follows: arrows leaving an action connect to IV objects created by the action while arrows entering an action connect IV objects to actions using them.

- Data Acquisition
- Data Curation
- Data Publishing
- Data Processing
- Data Use

Data Acquisition

Data Acquisition


«IV_DynamicSchemata»
Data Acquisition
«IV_Action»
Specify investigation design
«IV_Object»
specification_of_investigation_design :Metadata
state = raw
«IV_Action»
Specify measurements or observations
«IV_Object»
specification_of_measurement_or_observation :Metadata
state = raw
«IV_Action»
Perform measurements or Observations
«IV_Object»
measurement_result :PersistentData
state = raw
«IV_Action»
Store data
«IV_Object»
persistent_data :PersistentData
state = raw
«IV_Action»
Data Curation

Notation

The data acquisition phase encompasses the actions defined for the observation/experimentation, storage, identification and backup of measurements/observations (raw data).

The following paragraphs explain the detailed diagram of how the IV actions can be combined to support data acquisition.

> **Note**
> This example is provided for illustrative purposes. The example shows one of many alternatives for performing data acquisition. Other IV actions and IV objects can be introduced at this stage. Additional actions and objects not described in the IV of the ENVRI RM can also be incorporated.

Specify investigation design: Before a measurement or observation can be started the design (or setup) must be defined, including the working hypothesis and scientific question, method of the selection of sites (stratified / random), necessary precision of the observation or measurement, boundary conditions, etc. For correctly using the resulting data, details about their processing, and the parameters defined have to be available (e.g. if a stratified selection of sites according to parameter A is done, the resulting value of parameter A can not be evaluated in the same way as other results).

Specify measurement or observation: After defining the overall design of measurements or observations, the measurement method, complying with the design, including devices which should be used, standards / protocols which should be followed, and other details have to be specified. The details of the process and the parameters used have to preserved to guarantee correct interpretation of the resulting data (e.g. when modelling a dependency of parameter B of a parallel measured wind velocity, the limit of detection of the used anemometer influences the range of values of possible assertions).

Perform measurement or observation: After the measurement or observation method is defined, the experiment can be performed, producing measurement result(s) which is a form of persistent data in a raw state.

Store data: The measurement result data is stored.This action can be very simple when using a measurement device, which periodically sends the data to the data management system, but this can also be a sophisticated harvesting process or e.g. in case of biodiversity observations a process done by humans. The storage process is the first step in the lifecycle of data that makes data accessible in digital form.

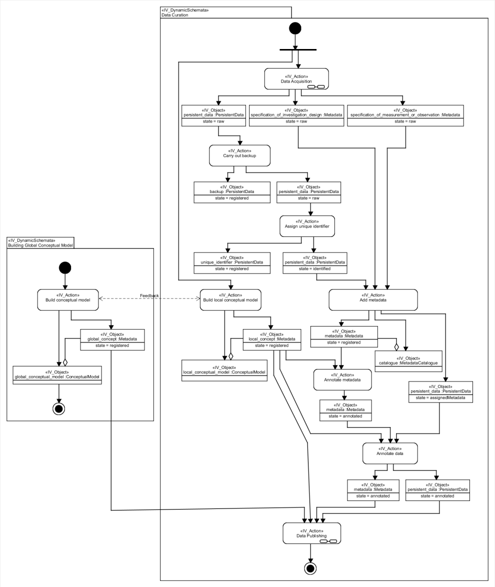
Data curation: Once data is stored, the next phase of the data lifecycle is data curation.

Data Curation

The data curation phase encompasses the actions that support the the long term preservation and use of research data. The main product of this set of actions is persistent data in a stable state (annotated data). The following paragraphs explain the detailed diagram of how the IV actions can be combined to support data curation.

Note
This example is provided for illustrative purposes. The example shows one of many alternatives for performing data curation. Other IV actions and IV objects can be introduced at this stage, for instance: Check quality, Register metadata, or Publish metadata. Actions and objects not described in the IV of the ENVRI RM can also be incorporated.

Data Curation



Notation

Data Acquisition: The first action is Data Acquisition, the phase of the data lifecycle that precedes data curation. This action produces three IV Objects: PersistentData, SpecificationOfMeasurementsOrObservations and SpecificationOfInvestigationDesign.

Carry out backup: As soon as data are available to the RI a backup can be made, independently of the state of the persisted data. This can be done locally or remotely, by the data owners or by dedicated data archiving centres.

Assign Unique Identifier: Data needs to be uniquely identified for correct retrieval and processing, the unique identifier can be local to the RI or global, to be used from outside the RI. As such it can be a simple numerical value assigned by the RI DBMS or a specific PID assigned following the standards of an external PID provider.

Add metadata: This action uses the specifications of investigation and measurements to facilitate the understanding of the associated persistent data object. In addition to this data the RI can add timestamps, and other identification data as metadata. Once the data is correctly stored and identified, and the corresponding metadata has been also created, persistent data can be linked to metadata.

Annotate data: Data is further enriched with additional metadata which can correspond to a specific ontology for the research field.

Annotate metadata: Metadata can also be further enriched with additional metadata which can correspond to a specific ontology for the research field.

Build conceptual model: The building of a **local conceptual model** mirrors the wider research community efforts to build a global conceptual model. In this set of activities concept are added to the local conceptual model of the RI. The conceptual model is made of the composition of concepts, which are used to help people know, understand, or simulate a subject the model represents. The pairing of data and metadata using semantic annotations creates a local concept (a new metadata object) and changes the state of the persistent data object to annotated.

Global conceptual models are ontologies, thesauri, dictionaries, or hierarchies built by a larger communities than a single RI, such as GEMET, DOLCE, SWEET. This action normally happens outside of the RI's main activities. Trough feedback mechanisms RIs participate in the creation of global conceptual models while developing their own models..

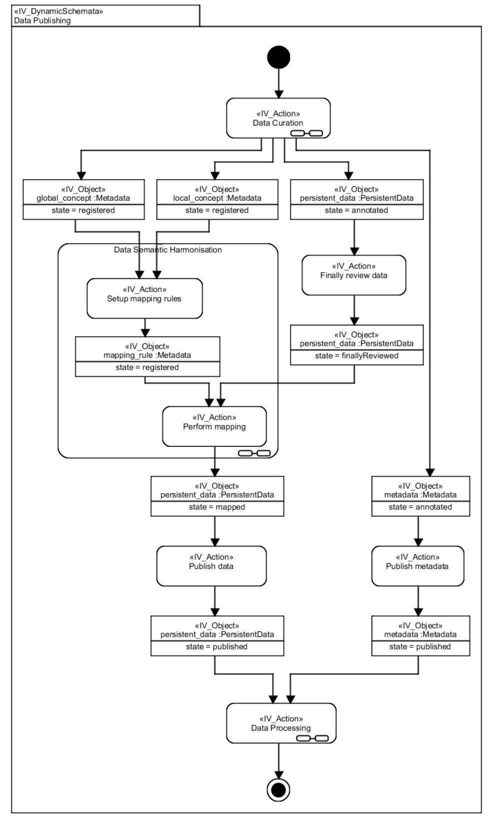
Data Publishing: Once data have been curated, the next phase of the data lifecycle is data publishing.

Data Publishing

The data publishing phase encompasses the actions that make the data available for entities (people and systems) outside the RI. The following paragraphs explain the detailed diagram of how the IV actions can be combined to support data curation.

> **Note**
> This example is provided for illustrative purposes. The example shows one of many alternatives for performing data curation. Other IV actions and IV objects can be introduced at this stage, for instance: QualityAssurance. Actions and objects not described in the IV of the ENVRI RM can also be incorporated.

Data Publishing



Notation

Data Curation: The first action is Data Curation, the phase of the data lifecycle that precedes data Publishing. This action produces four IV Objects: PersistentData, LocalConceptualModel, LocalConcept, and Metadata.

Finally Review Data: Persistent data that is in the process of publishing needs to be reviewed before proceeding to publising. It is important to clearly specify what the "finallyReviewed" state means. In some RIs it can mean, that those data will never change again, the optimum for the outside user. For other RIs it might also mean, that only under certain circumstances those data will be changed. In this case it is important to know what "certain circumstances" mean.

Build Global Conceptual Model: The construction of a global conceptual model makes sure that there is an appropriate fit between the persistent data to be published and their metadata (including the local conceptual model) with other models existing outside the RI. The GlobalConceptualModel is the representation of how that outside world looks to the RI.

Semantic Harmonisation: unifies data (and knowledge) models based on the consensus of collaborative domain experts to achieve better data (knowledge) reuse and semantic interoperability. This complex activity is performed in two stages: setup mapping rules and perform mapping, defined as follows.

Setup Mapping Rule: The Global model is used to generate a set of mapping rules to enable linking the RI data and metadata to global semantics. This may include simple conversions, such as conversions of units, but may also imply more sophisticated transformations like transformations of code lists, descriptions, measurement descriptions, and data provenance.

Perform mapping: This action carries out the linking of data and metadata to one or more global models.

Publish data: Mapped data is made available to the outside world. The PID is the main identifier of the data but the data can also be located by querying metadata.

Publish metadata: Metadata is also mapped and published to enable more sophisticated data querying.

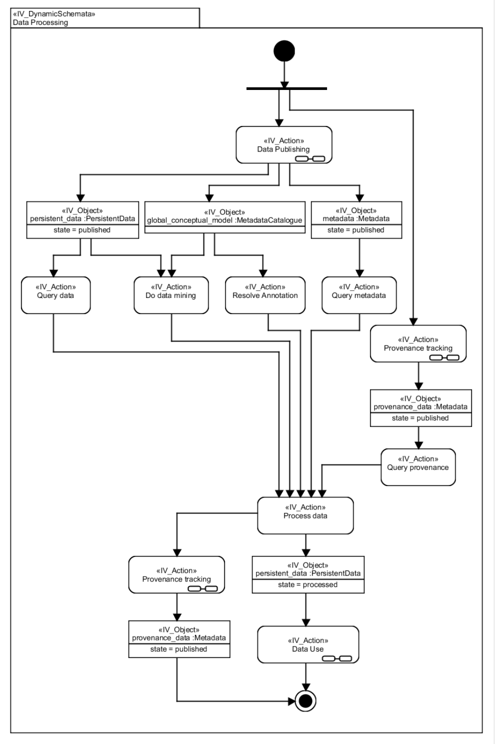
Data Processing: Once data have been published, the next phase of the data lifecycle is data processing.

Data can be made directly accessible or indirectly. Direct access means, that a data request to a data server (query data) gets the data or an error message as answer. Indirect access means, initially accessing metadata (query metadata), searching for a fitting data set and then querying on the resulting data set. Those two steps can be extended further, when intermediate steps are involved. The multi-step approach is often used for data, which are not open, making metadata open but not data itself. For queries touching several data sets and/or filtering the data (like e.g. give me all NOx air measurement where O3 exceeds a level of Y ppb) the multy-step approach can be seen blocker.

Data Processing

The data processing phase encompasses the actions that support making use of the RI published data. The following paragraphs explain the detailed diagram of how the IV actions can be combined to support data curation.

Data Processing


«IV_DynamicSchemata»
Data Processing
«IV_Action» Data Publishing
«IV_Object» persistent_data :PersistentData
state = published
«IV_Object» global_conceptual_model :MetadataCatalogue
«IV_Object» metadata :Metadata
state = published
«IV_Action» Query data
«IV_Action» Do data mining
«IV_Action» Resolve Annotation
«IV_Action» Query metadata
«IV_Action» Provenance tracking
«IV_Object» provenance_data :Metadata
state = published
«IV_Action» Query provenance
«IV_Action» Process data
«IV_Action» Provenance tracking
«IV_Object» persistent_data :PersistentData
state = processed
«IV_Object» provenance_data :Metadata
state = published
«IV_Action» Data Use

Notation

> **Note**
> This example is provided for illustrative purposes. The example shows one of many alternatives for performing data processing. Other IV actions and IV objects can be introduced at this stage. Actions and objects not described in the IV of the ENVRI RM can also be incorporated.

Data Publishing: The first action is Data publishing, the phase of the data lifecycle that precedes Data Processing. This action produces three IV Objects: PersistentData, GlobalConceptualModel, and Metadata which are used in the data processing phase to access and process data.

Provenance Tracking: Is the action that keeps a log about the the actions and the data state changes as data evolves through the RI systems. The resulting provenance data is a form of metadata which may be of interest for referencing and citing the use of data within and outside the RI.

Query data: This action requests specific persisted data from the RI.

Do data mining: This action implies the execution of a sequence of metadata/data request/interpret/result/request which automatically produce or find patterns in the data being analysed. Usually this sequence helps to deepen the knowledge about the data.

Resolve annotation: This action implies finding a specific data set from a set of semantic annotation and constrains on those annotations. If the annotation is resolved the result should be a link or a set of links to specific data sets, if not the result is an empty set.

Query metadata: This action requests specific persisted data from the RI using metadata as additional parameters for narrowing down the search.

Query provenance: This action requests specific persisted data about the provenance of some data or metadata. This is usually done to determine the origin and validity of data but can also be helpful for citation and referencing.

Process data: The performance of any of the five actions listed before, is automatically detected as a form of data processing by the RI system. This should result in changing the state of the data to "processed". The processed state can mean several things such as: the data has been consulted, the data has been referenced, the data has been downloaded, the data has been used as input for an external process, etc.

Data use: Once data have been processed, the next phase of the data lifecycle is data use, which to some extent overlaps with processing.

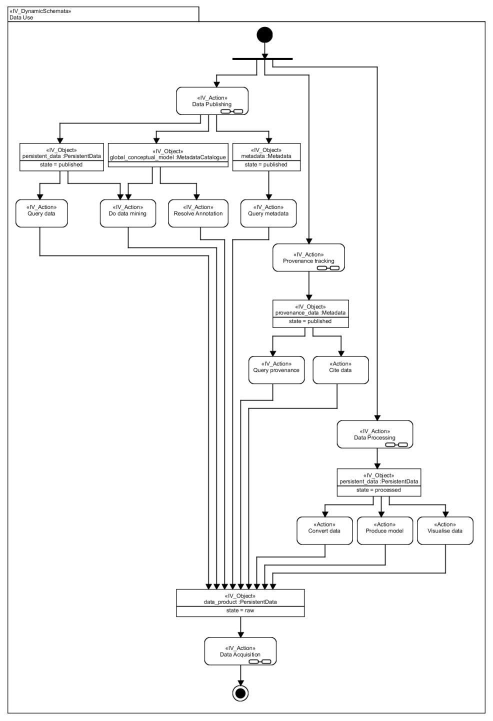
Provenance Tracking: As described in the overview, the provenance tracking action tracks all changes in the states of persistent data. This is an important action which has wide use inside and outside the RI.

Data Use

The data use phase is a bridge phase which sits between processing and acquisition. In this phase, the data is used and may produce new data (raw data) which can in turn be persisted by an RI. The actions that act on data at this point can be provided by same RI exposing the data or by external entities (RIs or other).

In the use phase, the RI system is open to the outside world. Users (persons or external systems) can use the services provided to produce new data products.

Data Use



Notation

Note
This example is provided for illustrative purposes. The example shows one of many alternatives for performing data processing. Other IV actions and IV objects can be introduced at this stage. Actions and objects not described in the IV of the ENVRI RM can also be incorporated. In the diagram and associated descriptions below, cite data, convert data, produce model, and visualise data are some examples of these types of actions.

Data Publishing: The first action is Data publishing, the phase of the data lifecycle that precedes data Processing. This action produces three IV Objects: PersistentData, GlobalConceptualModel, and Metadata which are used in the Data Use phase to access and process data.

Provenance Tracking: Provenance tracking keeps a log about the the actions and the data state changes as data evolves through the RI systems. The resulting provenance data is a form of metadata which may be of interest for referencing and citing the use of data within and outside the RI.

Data Processing: Data Processing produces Persistent Data IV Objects.

Query data: This action requests specific persisted data from the RI.

Do data mining: This action implies the execution of a sequence of metadata/data request/interpret/result/request which automatically produce or find patterns in the data being analysed. Usually this sequence helps to deepen the knowledge about the data.

Resolve annotation: This action implies finding a specific data set from a set of semantic annotation and constrains on those annotations. If the annotation is resolved the result should be a link or a set of links to specific data sets, if not the result is an empty set.

Query metadata: This action requests specific persisted data from the RI using metadata as additional parameters for narrowing down the search.

Query provenance: This action requests specific persisted data about the provenance of some data or metadata. This is usually done to determine the origin and validity of data but can also be helpful for citation and referencing.

Cite data: Produce a reference to persistent data or metadata.

Convert data: converting and generating data products, for instance translating to a different format.

Produce model: creation of statistical models, simulation models or summaries with the data provided.

Visualise data: creating visual models which display data alpha-numerically, graphically, or geographically.

Data Acquisition: Use of data has the potential for creating data products which may need to be persisted, re-initiating the data lifecycle. For this reason, the the last action after Data Use actions is Data Acquisition.

IV Information Management Constraints

The IV of the ENVRI RM provides the means for specifying constrains which describe the set of rules governing data management. The set of models used for describing constraints are part of the static schemata in ODP [37]. In the ENVRI RM information management constraints establish mechanisms to:

1. avoid loss of data around measurements and observations.
2. provide information about the meaning of data.
3. make data and metadata available for external use.

The IV of the ENVRI RM provides three types of management constraints:

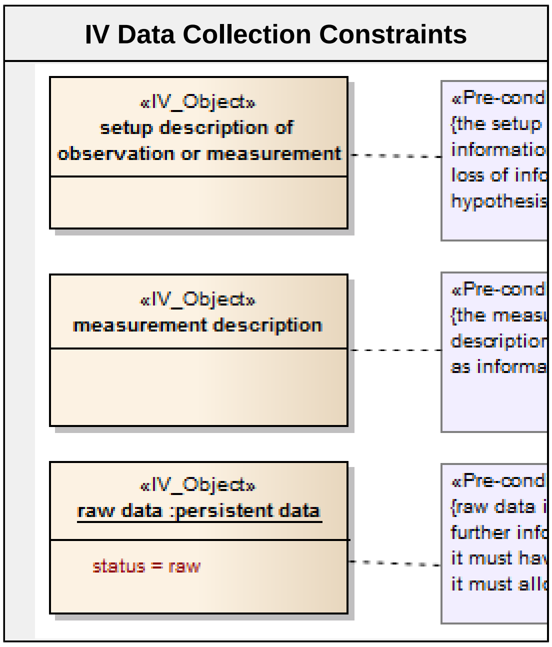
- Data Collection Constraints
- Data Integration Constraints
- Data Publication Constraints

Data Collection Constraints

The constraints applied to data collection are illustrated in the figure below. The application of these constrains helps to avoid data loss or wrong interpretation.

Observing these three rules together ensures that data can be correctly interpreted and reduces the risk of data loss. This is because the rules guarantee that the original data can be retrieved and interpreted correctly though the lifetime of the information objects derived from them.

The rules guarantee the availability of the rationale for collecting 1st rule), the details about the how collection proceeded (2nd rule), and the original data collected.



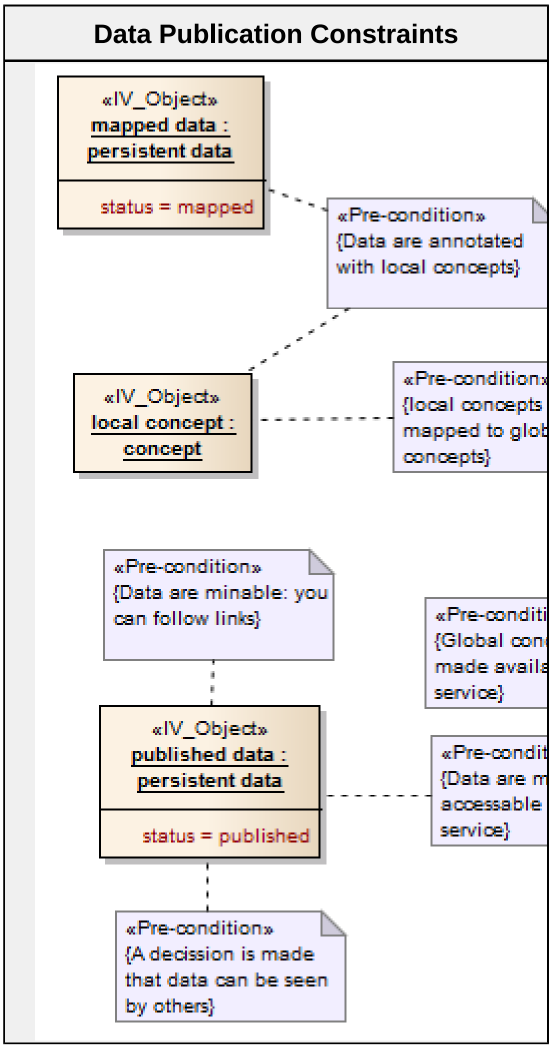
Data Integration Constraints

The constraints for data integration are illustrated in the following figure. Data integration constraints support the correct interpretation of data, helping external data users correctly interpret and map the semantics of data.

The observation of these rules makes possible integrating data within a RI and the outside world. This requires adding a special type of metadata, a local conceptual model and then mapping the data to a global conceptual model. Mapping data to global semantics may include simple tasks such as conversions of units, but can also need sophisticated transformations such as code lists cross-referencing or measurement descriptions, and data provenance.

Data Publication Constraints

Constraints for data publication are illustrated in the following figure. The constrains specify conditions necessary for preparing the data to be publicly accessed.


Data Publication Constraints
«IV_Object»
mapped data : persistent data
status = mapped
«Pre-condition»
{Data are annotated with local concepts}
«IV_Object»
local concept : concept
«Pre-condition»
{local concepts mapped to glob
concepts}
«Pre-condition»
{Data are minable: you can follow links}
«Pre-conditi
{Global con
made availa
service}
«IV_Object»
published data : persistent data
status = published
«Pre-condit
{Data are m
accessable
service}
«Pre-condition»
{A decission is made that data can be seen by others}

Computational Viewpoint

A research infrastructure (RI) provides a context in which investigators can interact with scientific data in a principled manner. To provide this context, an RI must support a portfolio of possible research interactions. These interactions can be realised by binding together different services via standard operational interfaces.

The Computational Viewpoint (CV) accounts for the major computational objects that can be found within an environmental research infrastructure, as well as the interfaces by which they can be invoked, and by which they can invoke other objects in the infrastructure. Each object encapsulates functionality that should be implemented by a service or other resource in a compliant RI. Binding of computational objects together via compatible interfaces creates a network of interactions that allows an RI to support the data related activities of its target research community.

The description of the CV is divided in three parts: **objects** , **CV Objects and Subsystems**, and **CV Integration points**.

- **CV Objects**: present computational objects according a generic architecture of the RIs.
- **CV Objects and Subsystems**: presents examples of how components are integrated for supporting the data lifecycle into five different subsystems.
- **CV Integration points**: defined to support the movement of research data between phases.

Note

The Computational Viewpoint defines computational objects (CV Object) and interfaces (CV Interfaces) which enable their interaction.

The diagram below shows the main elements of the CV and their relationships. Each ellipse contains a concept. The arrows connecting the concepts are directed and indicate the relationship between concepts. The label of the link indicates the type of relationship. From this, the diagram indicates that a CV object provides a CV interface, as indicated by the **provides** relationship. Similarly, a CV object can create another CV object, as indicated by the **canInstantiate** relationship. In this same way a CV interface can fit another CV interface, this is indicated the **fits** relationship.

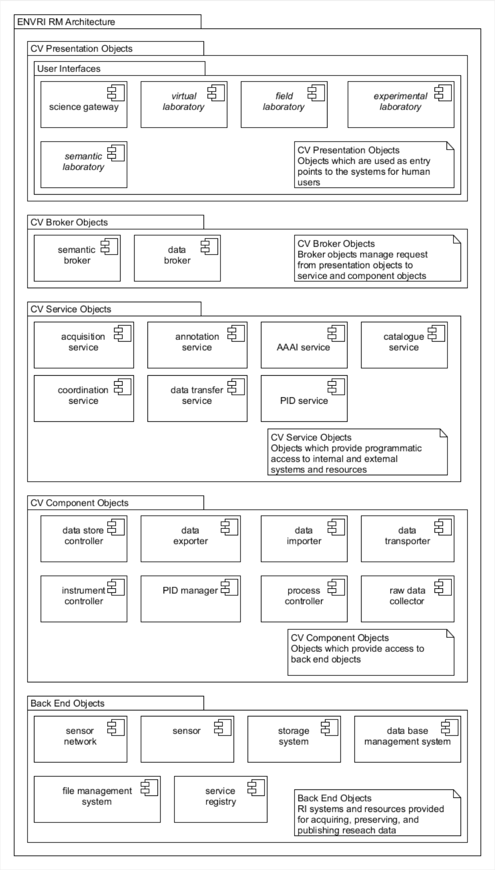
Computation Viewpoint components and their relationships

Before proceeding, the reader may wish to study the pages on How to read the Model (Computational Viewpoint)and How to use the Model (Computational Viewpoint).

CV Objects

The archetype of a modern environmental research infrastructure has a brokered, service-oriented architecture. Core functionality is encapsulated within a number of key resources which can be accessed by means of externally-facing gateway services. Interaction by external agents with internal resources is overseen by one or more brokers (often closely integrated with the gateway) charged with validating requests and providing, where needed, an interoperability layer between otherwise heterogeneous components. The Computational Viewpoint of the ENVRI RM provides a set of models which can help in the design, implementation, maintenance, and evolution of the systems and services that RIs provide for accessing data.

CV Objects



Notation

The CV prescribes a number of types of computational object for which there should be instances present in or around a research infrastructure in order to ensure that particular **Appendix A Common Requirements of Environmental Research Infrastructures** are supported. The grouping of CV objects into sets is based on the software architecture that is expected to be implemented when providing access to data and other related resources during the research **Model Overview** (evident in the RIs analysed as part of the **The ENVRI and ENVRIplus Projects**) projects. Consequently, the presentation of CV objects is arranged as five sets corresponding to each of the architectural layers of RI systems (note however, that this should not be read as prescriptive and that other groupings are possible):

- **CV Presentation Objects:** computational objects that facilitate access to RIs by human users.
- **CV Broker Objects:** computational objects that act as intermediaries for access to data held within the data store and facilitate performing semantic interpretation and routing of queries.
- **CV Service Objects:** computational objects that offer programmatic access to distributed systems and resources (internal and external).
- **CV Component Objects:** computational objects that provide access to back end objects.
- **Back End Objects:** computational objects that encompass the RI's systems and resources for accessing research data and derived data products.

The set of CV components included in the ENVRI RM comprises the computational objects that are common to many RIs. The set is not closed, so each RI can include the additional components they require to completely model their systems.The set does not contain compulsory items, so each RI can exclude objects and interfaces that are not relevant for them.

Note
Before proceeding, the reader may wish to study the pages on How to read the Model (Computational Viewpoint)and How to use the Model (Computational Viewpoint).

CV Presentation Objects

CV Presentation objects are the entry points for human users to the systems and services provided to access research data and their derived products.

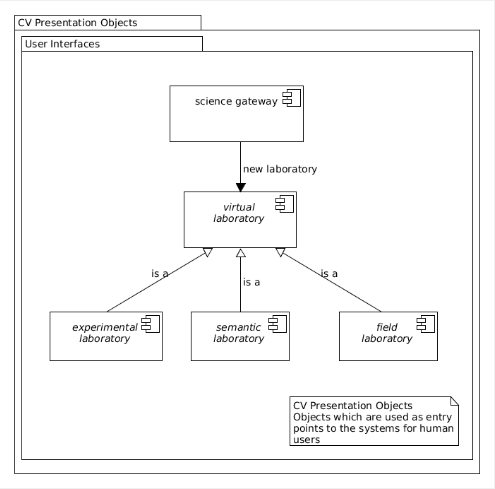
In the ENVRI RM, complex interactions between the components facilitating data use and other components are mediated by **virtual laboratory**; these objects are deployed by **science gateways** in order to provide a persistent context for such interactions between groups of users and components within the RI.

The Reference Model recognises the following specific sub-classes of laboratory:

- **field laboratory** (so-named because they interact with raw data sources 'in the field') are used to interact with the **CV Data Acquisition** components, allowing researchers to deploy, calibrate and un-deploy instruments as part of the integrated data acquisition network used by an infrastructure to collect its primary 'raw' data. Field laboratories have the ability to instantiate new **instrument controller** from the data acquisition set.
- **experiment laboratory** are used to interact both with curated data and data processing facilities, allowing researchers to deploy datasets for processing and acquire results from computational experimentation.
- **semantic laboratory** are used to interact with the semantic models used by a research infrastructure to interpret datasets and characteristic (meta)data.

Regardless of provenance, all laboratories must interact with a **AAAI service** in order to authorise requests and authenticate users of the laboratory before they can proceed with any privileged activities.

CV Presentation Objects



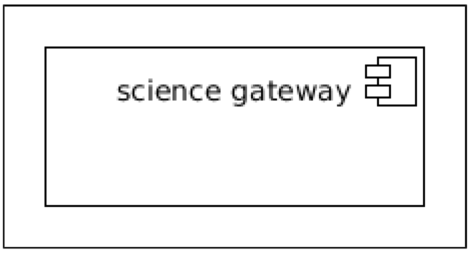
Notation

Science gateway

Community portal for interacting with an infrastructure.

A science gateway object encapsulates the functions required to interact with a research infrastructure from outside with the objects provided for data acquisition, data curation, data brokering and data processing. A science gateway should be able to provide virtual 'laboratories' for authorised agents to interact with and possibly configure many of the science functions of a research infrastructure. A science gateway is also known as a Virtual Research Environment.

- A science gateway object can instantiate any number of **virtual laboratory** objects.

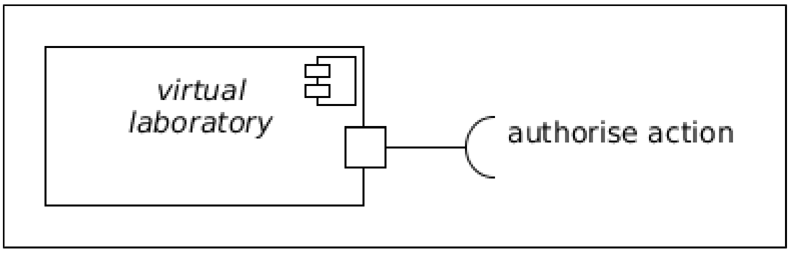


Virtual laboratory

Community proxy for interacting with RI components.

A virtual laboratory object encapsulates interaction between a user or group of users and a subset of the science functions provided by a research infrastructure. Its role is to bind a **AAAI service** with (potentially) any number of other infrastructure objects.

A virtual laboratory object must provide at least one interface:

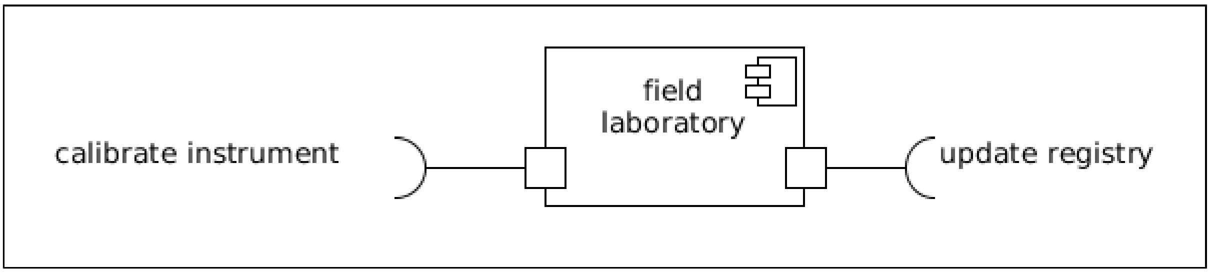


- **authorise action (client)** is used to retrieve authorisation for any restricted interactions with the data acquisition components.

Specific sub-classes of virtual laboratory should be defined to interact with the infrastructure in different ways. The ENVRI RM defines the **field laboratory** object for interaction with the **CV Data Acquisition** components.

Field laboratory

Community proxy for interacting with



data acquisition instruments.

A sub-class of **virtual laboratory** object encapsulating the functions required to access, calibrate, deploy or withhold instruments during the data acquisition phase.

Deployment of an instrument entails the deployment of an instrument controller by which the instrument can be interacted with.

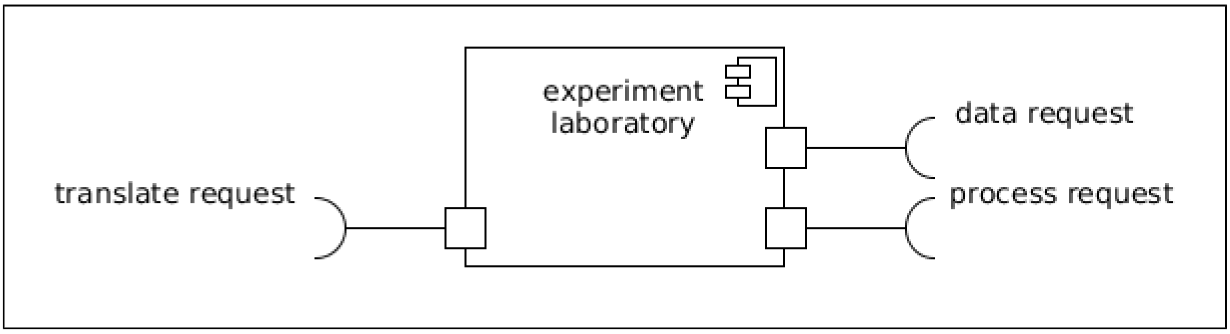
- A field laboratory object can instantiate any number of **instrument controller** objects.

A field laboratory should provide at least two operational interfaces in addition to those provided by any virtual laboratory:

- **calibrate instrument (client)** is used to calibrate the reading of data by instruments based (in principle) on scientific analysis of data output. This interface can also be used to monitor activity on a given instrument.
- **update registry (client)** is used to register and/or withdraw instruments used for data acquisition.

> A field laboratory is created by a science gateway in order to allow researchers in the field to interact with the data acquisition objects.
>
> The degree of freedom with which a field laboratory interacts with other data acquisition objects is contingent on the nature of the research infrastructure and policed by a **AAAI service** object (as defined for all user laboratories).

Experiment laboratory



Community proxy for conducting experiments within a research infrastructure.

A sub-class of **virtual_laboratory** object encapsulating the functions required to schedule the processing of curated and user-provided data in order to perform some task (analysis, data mining, modelling, simulation, etc.).

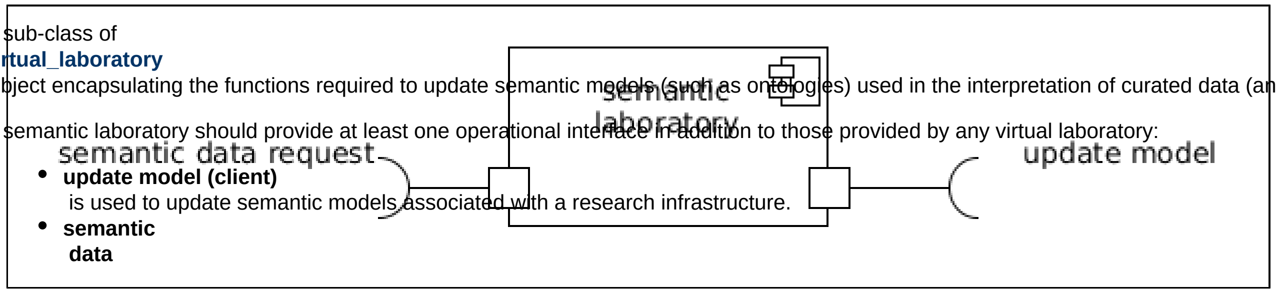
An experiment laboratory should provide at least three operational interfaces:

- **data request (client)** is used to make requests of the research infrastructure pertaining to curated datasets.
- **process request (client)** is used to make requests of the research infrastructure pertaining to data processing.
- **translate request (client)** is used to invoke a semantic broker where some mapping between different semantic domains is deemed necessary.

> An experiment laboratory is created by a science gateway to allow researchers interaction with data held by a research infrastructure in order to achieve some scientific output.

Semantic laboratory

Community proxy for interacting with semantic models.



A sub-class of **virtual_laboratory** object encapsulating the functions required to update semantic models (such as ontologies) used in the interpretation of curated data (and

A semantic laboratory should provide at least one operational interface in addition to those provided by any virtual laboratory:

- **update model (client)** is used to update semantic models associated with a research infrastructure.
- **semantic data request** (client) is used to make requests of the research infrastructure about metadata and annotations referring to data stored by the data

> A semantic laboratory is created by a science gateway in order to allow researchers to provide input on the interpretation of data gathered by a research infrastructure.

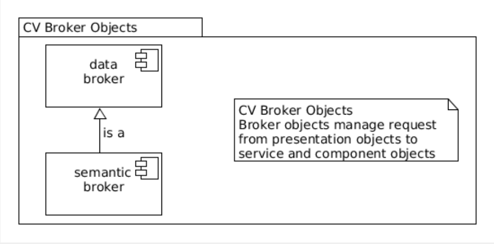
CV Broker Objects

Broker objects act as intermediaries for access to data held within the data store and facilitate performing semantic interpretation and routing of queries. For this brokers keep registries of **CV Service Objects** to which actions are routed. Whenever possible, advanced brokers which make use of metadata should be preferred to hard coded brokers.

- **data broker** objects act as intermediaries for access to data held within the data store.
- **semantic broker** objects perform semantic interpretation.

Brokers are responsible for verifying the agents making access requests and for validating those requests. These brokers can be interacted with directly via **virtual laboratory** such as **experiment laboratory** (for general interaction with data and processing services) and **semantic laboratory** (by which the community can update semantic models associated with the research infrastructure).

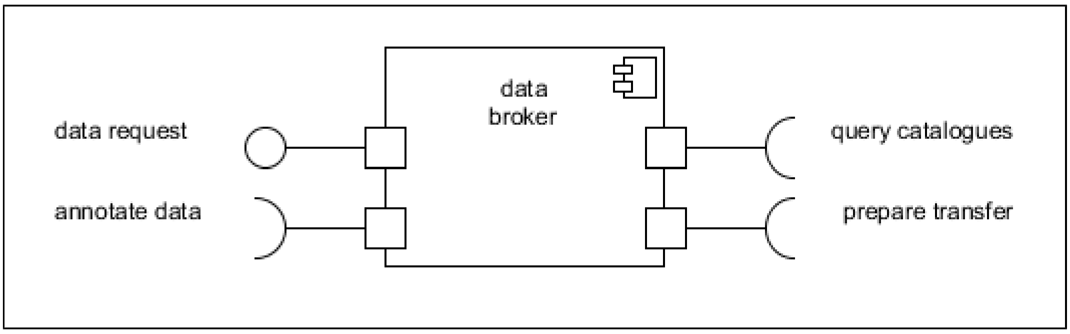
CV Broker Objects


CV Broker Objects
data broker
is a
semantic broker
CV Broker Objects
Broker objects manage request from presentation objects to service and component objects

Notation

Data broker

Broker for facilitating data access/upload requests.



A data broker object intercedes between the data publishing objects and the data curation objects, collecting the computational functions required to negotiate data transfer and query requests directed at data curation services on behalf of some user. It is the responsibility of the data broker to validate all requests and to verify the identity and access privileges of agents making requests. It is not permitted for an outside agency or service to access the data stores within a research infrastructure by any means other than via a data broker.

A data broker should provide four operational interfaces:

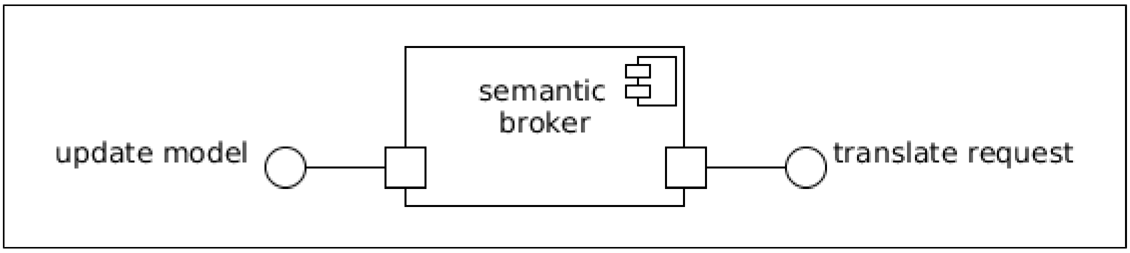
- **data request (server)** provides functions for requesting the import or export of datasets, the querying of data or the annotation of data within a research infrastructure.
- **annotate data (client)** is used to request annotation of data held within the data curation objects of a research infrastructure.
- **prepare data transfer (client)** is used to negotiate data transfers with the data curation objects of a research infrastructure.
- **query catalogues (client)** is used to forward queries onto the data curation objects of a research infrastructure and receive the results.

> Data brokers are not responsible for brokering the collection of raw data from the data acquisition objects, as this is handled

more efficiently by an acquisition service.

Semantic broker

Broker for establishing semantic links between concepts and bridging queries between semantic domains.



A semantic broker intercedes where queries within one semantic domain need to be translated into another to be able to interact with curated data. It also provides the functionalities required to update the semantic models used by an infrastructure to describe data held within.

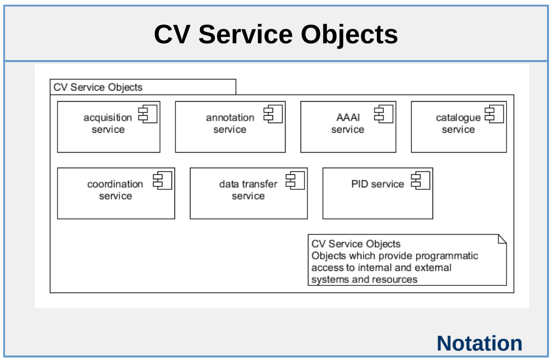
A semantic broker should provide two operational interfaces:

- **translate request (server)** provides functions for translating requests between two semantic domains.
- **update model (server)** provides functions for updating semantic models associated with a research infrastructure.

CV Service Objects

CV service objects offer programmatic access to distributed systems and resources (internal and external). This allows building RIs using both internal and external sourced components. The service layer includes the main services that enable data access, processing and transformation used in different phases of the research data lifecycle.

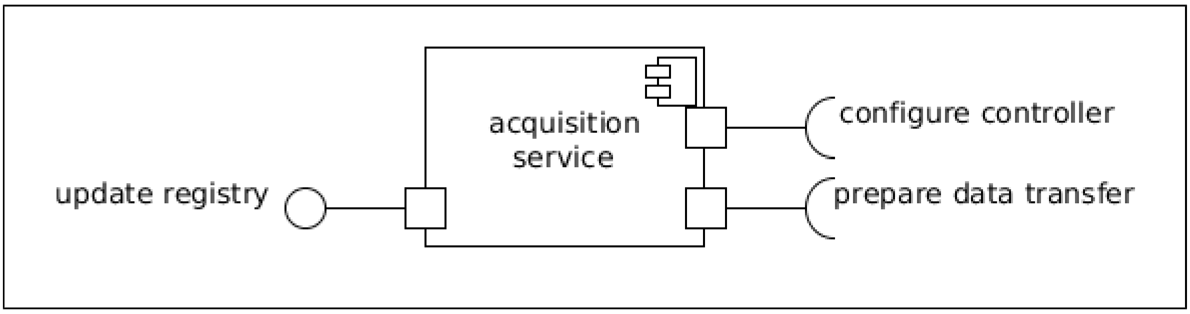
- The **acquisition services** , responsible for ensuring that any data is delivered into the infrastructure in accordance with current policies.
- The **annotation service**, concerned with the updating of records (such as datasets) and catalogues in response to user annotation requests.
- The **AAAI service** handles authorisation requests and authentication of users before they can proceed with any privileged activities.
- The **catalogue service**, concerned with the cataloguing of metadata and other characteristic data associated with datasets stored within the infrastructure.
- The **coordination service** delegates all processing tasks sent to particular execution resources, coordinates multi-stage workflows and initiates execution.
- The **data transfer service**, concerned with the movement of data into and out of the infrastructure.
- The   **PID service** provides globally-readable persistent identifiers (PIDs) to infrastructure entities, mainly datasets, that may be cited by the community.



Acquisition service

Oversight service for integrated data acquisition.

An acquisition service object encapsulates the computational functions



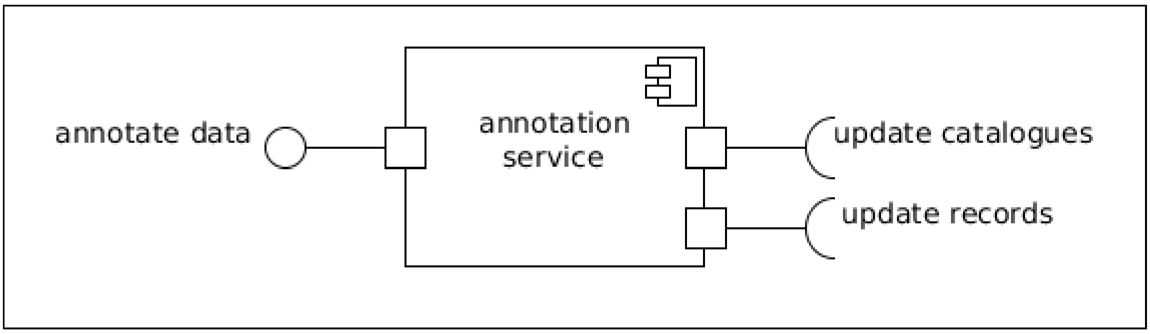
required
to monitor and manage a network of instruments. An acquisition service can translate acquisition requests into sets of individual instrument configuration operations as appropriate.

An acquisition service should provide at least three operational interfaces:

- **update registry (server)** provides functions for registering and deregistering instruments within the data acquisition phase.
- **configure controller (client)** is used to configure data collection (and other configurable factors) on individual instruments.
- **prepare data transfer (client)** is used to negotiate data transfers to data curation objects.

Annotation service

Oversight service for adding and updating records attached to curated datasets.



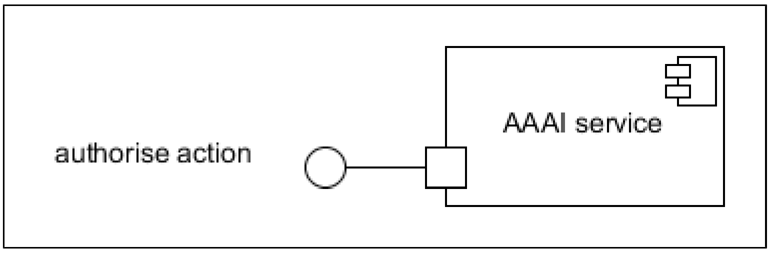
An annotation service object collects the functions required to annotate datasets and collect observations that can be associated with the various types of data managed within a research infrastructure.

An annotation service should provide three operational interfaces:

- **annotate data (server)** provides functions for requesting the annotation of existing datasets or the creation of additional records (such as qualitative observations made by researchers).
- **update catalogues (client)** is used to update catalogues or catalogue information managed by a catalogue service.
- **update records (client)** is used to update annotation records of existing datasets curated within one or more data stores.

AAAI service

Oversight service for authentication, authorisation, and accounting of user requests to the infrastructure.



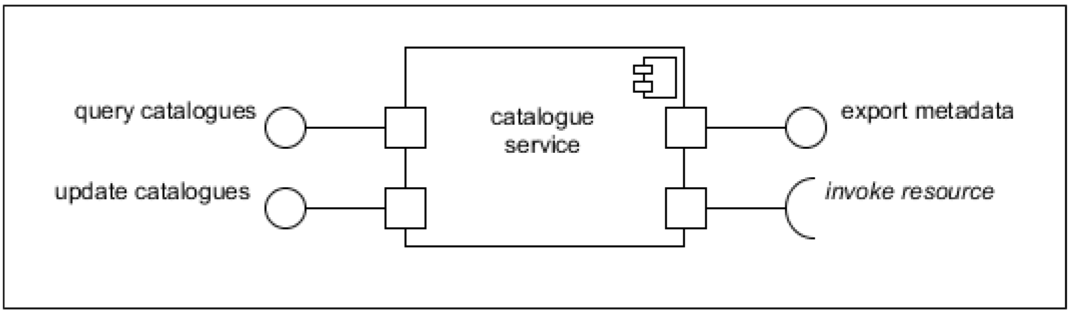
An AAAI service object encapsulates the functions required to authenticate agents, authorise any requests they make to services within a research infrastructure, and track their actions. Generally, any interaction occurring via a science gateway object or a virtual laboratory object will only proceed after a suitable transaction with an AAAI service object has been made.

An AAAI service should provide at least one operational interface:

- **authorise action (server)** provides functions to verify and validate proposed actions, providing authorisation tokens (for example) where required

Catalogue service

Oversight service for cataloguing curated datasets.



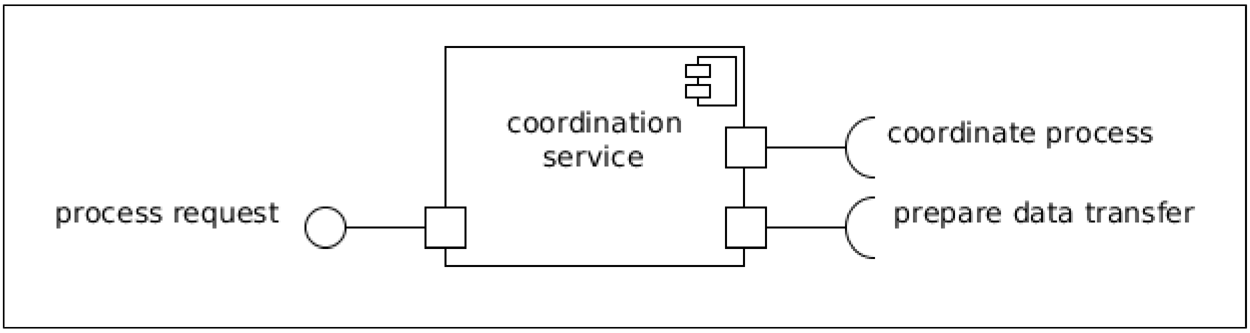
A catalogue service object collects the functions required to manage the construction and maintenance of catalogues of metadata or other characteristic data associated with datasets (including provenance and persistent identifiers) stored within data stores registered.

A catalogue service should provide four operational interfaces:

- **export metadata (server)** provides functions for gathering metadata to be exported with datasets extracted from the data curation store objects (data stores).
- **query catalogues (server)** provides functions for querying data held by the infrastructure, including the retrieval of datasets associated with a given persistent identifier.
- **update catalogues (server)** provided to update data catalogues metadata and their associated data assets.
- ***invoke resource (client)*** a generic interface provided 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.

> A data catalogue is itself a dataset, and can therefore be accessed and queried exactly as any other dataset.

Coordination service

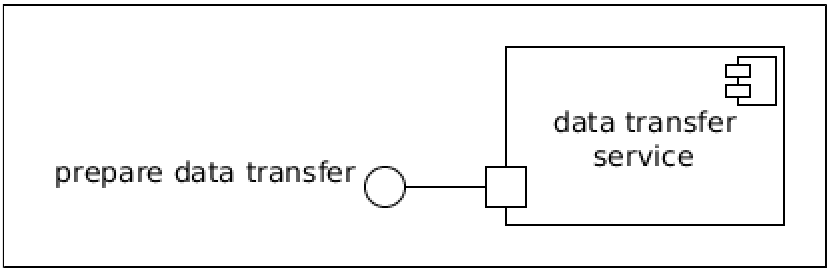


Oversight service for data processing tasks deployed on infrastructure execution resources.

A coordination service should provide at least three operational interfaces:

- **process request (server)** provides functions for scheduling the execution of data processing tasks. This could require executing complex workflows involving many (parallel) sub-tasks.
- **coordinate process (client)** is used to coordinate the execution of data processing tasks on execution resources presented by process controllers.
- **prepare data transfer (client)** is used to move data into and out of the data store objects in order to register new results or in preparation for the generation of such results.

Data transfer service



Oversight service for the transfer of data into and out of the data store objects.

A data transfer service object encapsulates the functions required to integrate new data into the RI and export that integrated data on demand. The data transfer service is responsible for setting up data transfers, including any repackaging of datasets necessary prior to delivery.

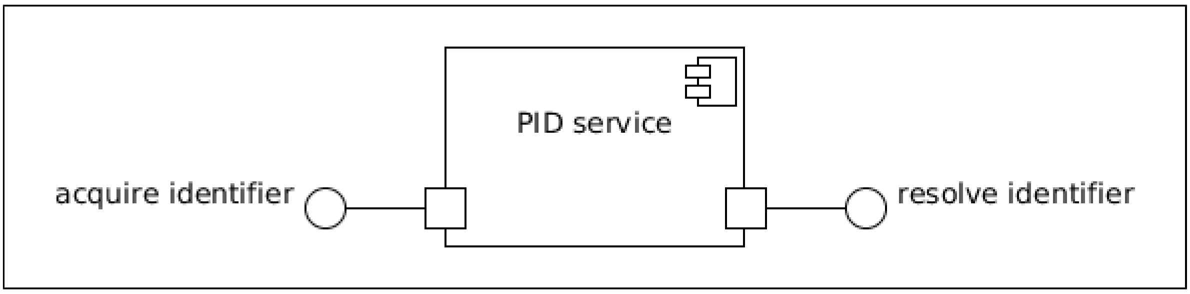
A data transfer object can create any number of new **data transporter** objects.

A data transfer service should provide one operational interface:

- **prepare data transfer (server)** provides functions for negotiating and scheduling a data transfer either into or out of the data stores of a RI.

> The actual coordination of data transfers is handled by data transporter objects; the data transfer service is responsible for specifying the behaviour of a given transporter.

PID service



External service for persistent identifier assignment and resolution.

Persistent identifiers are generated by a global service generally provided by an outside entity supported by the research community. A PID (persistent identifier) service object encapsulates this service and is responsible for providing identifiers for all entities that require them.

A PID service should provide at least two operational interfaces:

- **acquire identifier (server)** provides a persistent identifier for a given entity.
- **resolve identifier (server)** resolves identifiers, referring agents to the identified entity (in practice a science gateway providing access to the entity).

> Different versions of artefacts, where maintained separately, are assumed to have different identifiers, but those identifiers can share a common root such that the family of versions of a given artefact can be retrieved in one transaction, or only the most recent (or otherwise dominant) version is returned.

CV Component Objects

CV Component Objects

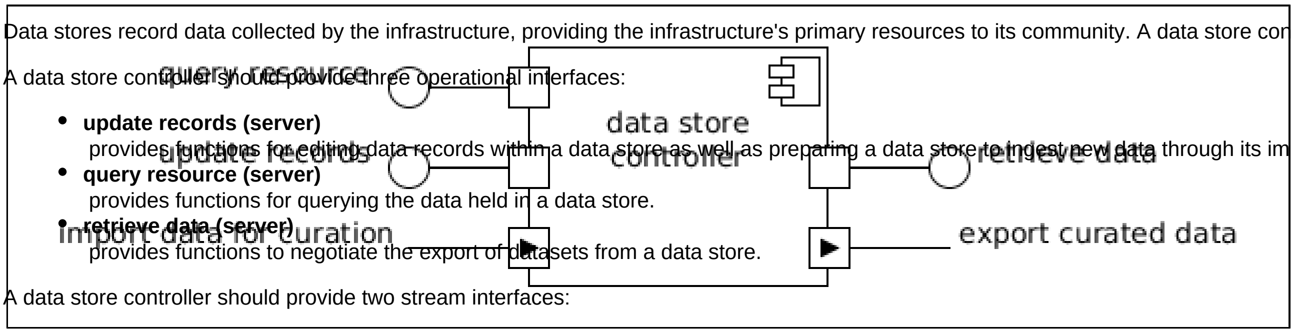
Notation

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 contro

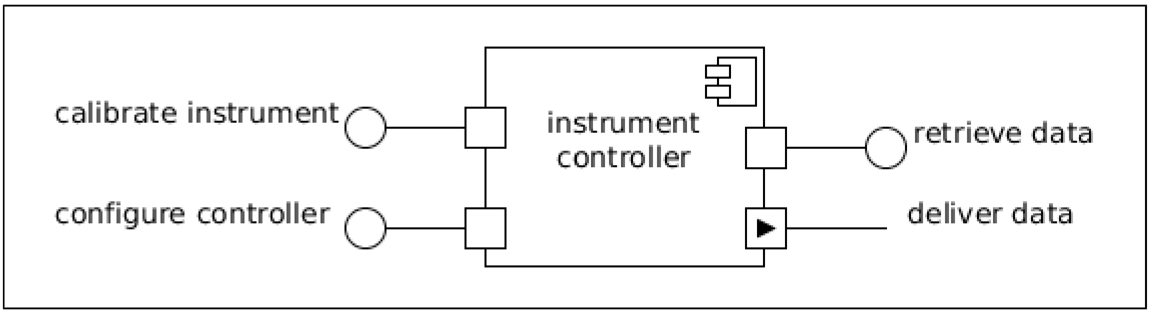
A data store controller should provide three operational interfaces:

- **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 impo
- **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:

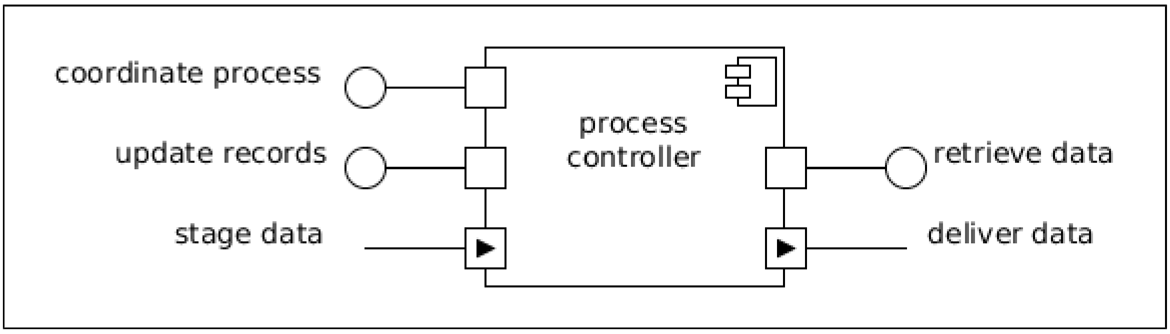
- **calibrate instrument (server)** provides functions to calibrate the reading of data by an instrument (if possible).
- **configure controller (server)** provides functions to configure how and when an instrument delivers data to a data store.
- **retrieve data (server)** provides functions to directly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

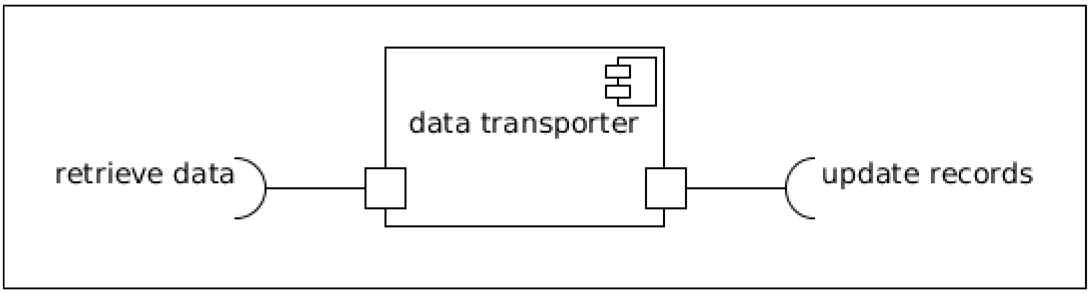
A process controller should provide at least three operational interfaces:

- **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



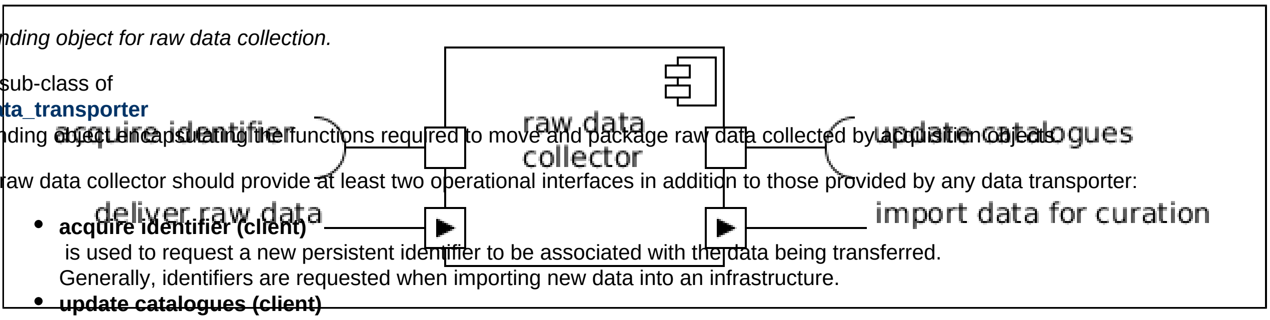
Generic binding object for data transfer interactions.

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.

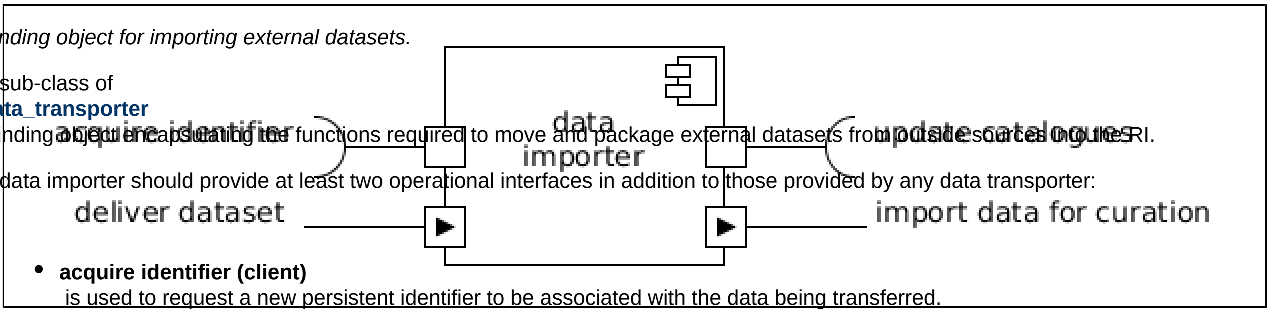
A raw data collector should provide at least two operational interfaces in addition to those provided by any data transporter:

- **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 ne

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:

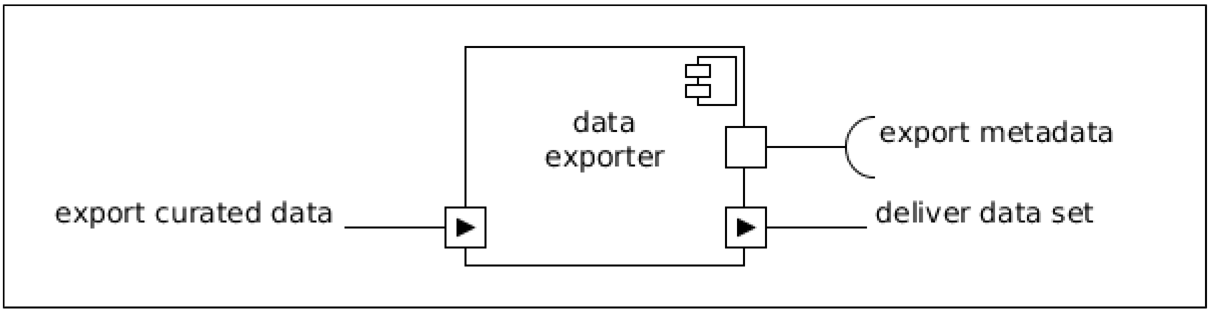
- **acquire identifier (client)** is used to request a new persistent identifier to be associated with the data being transferred. Generally, identifiers are requested when importing new data into an infrastructure.

- **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 sub-class of **data_transporter** binding object encapsulating the functions required to move and package curated datasets from the data curation objects to an outside destination.

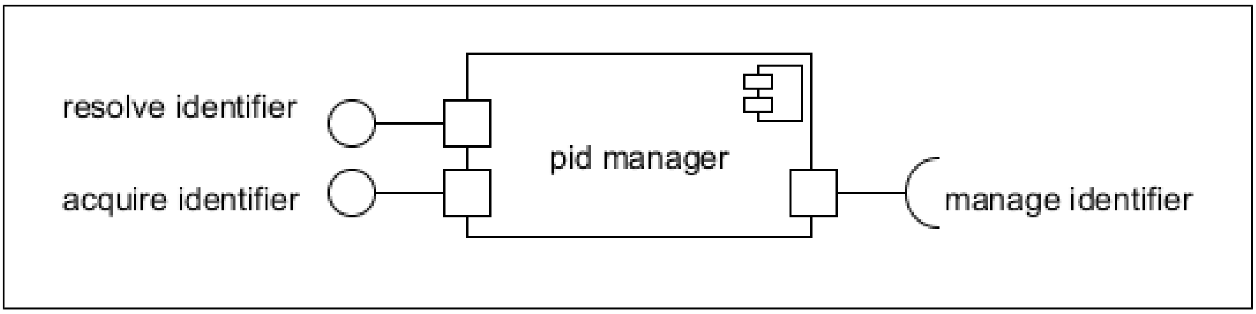
A data exporter should provide at least one operational interface in addition to those provided by any data transporter:

- **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



An object 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 identifer (client)** function for retrieving, updating and deleting identifiers interacting with a pid service.

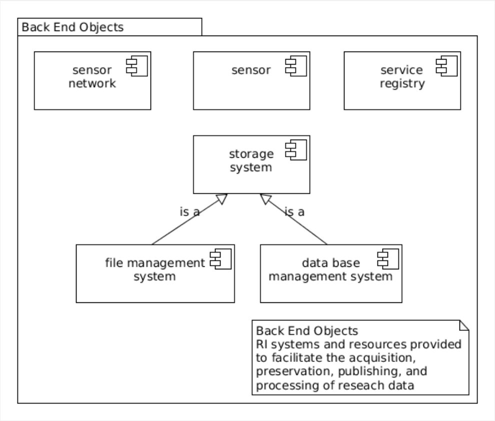
CV Back End Objects

Back End Objects are computational objects which encompass the RI's systems and resources provided for acquiring, preserving, publishing, and processing research data and derived data products.

- **Sensor network:** is a network consisting of distributed sensors which monitor physical or environmental conditions.

- **Sensor**: is a converter that measures a physical quantity and converts it into a signal which can be read by an observer or by an (electronic) instrument.
- **Storage System**: is a systems that manages the storage and retrieval of data and metadata.
- **File Management System**: is a storage systems that manages the storage and retrieval of data as files in a computer system.
- **Database Management System**: is a storage systems that manages the storage and retrieval of data and metadata into logically structured repositories.
- **Service Registry**: is an information system for registering services.

CV Back End Objects


Back End Objects
sensor network
sensor
service registry
storage system
is a
is a
file management system
data base management system
Back End Objects
RI systems and resources provided to facilitate the acquisition, preservation, publishing, and processing of reseach data

Notation

CV Objects and Subsystems

The **SV Community Roles** include five subsystem roles which support each of the phases of the data lifecycle. In this section, the models of those subsystems are developed further using computational viewpoint components. The five subsystems defined are:

- **CV Data Acquisition**
- **CV Data Curation**
- **CV Data Publishing**
- **CV Data Processing**
- **CV Data Use**

CV RI subsystems

Notation

Note
Before proceeding, the reader may wish to study the pages on How to read the Model (Computational Viewpoint)and How to use the Model (Computational Viewpoint).

CV Data Acquisition

The basis for environmental research is the observation and measurement of environmental phenomena. The archetypical environmental research infrastructure provides access to data harvested from an extended network of sensors, instruments and other contributors deployed in the field. The following examples present the acquisition of data from instruments and from external data sources.

Data acquisition from sensors

The diagram shows the organisation of five CV objects as part of an RI which are used for collecting data from an instrument. The instrument controller could be a simple device collecting data from a single sensor or a complex device managing the collection of data for a sensor network.

Data Acquisition Subsystem - collect data

Notation

Acquisition is manipulated via **field laboratory**, community proxies by which authorised agents can add and remove instruments from the network (by registering and de-registering instrument controllers) as well as calibrate instrument readings where applicable in accordance with current community best-practice.

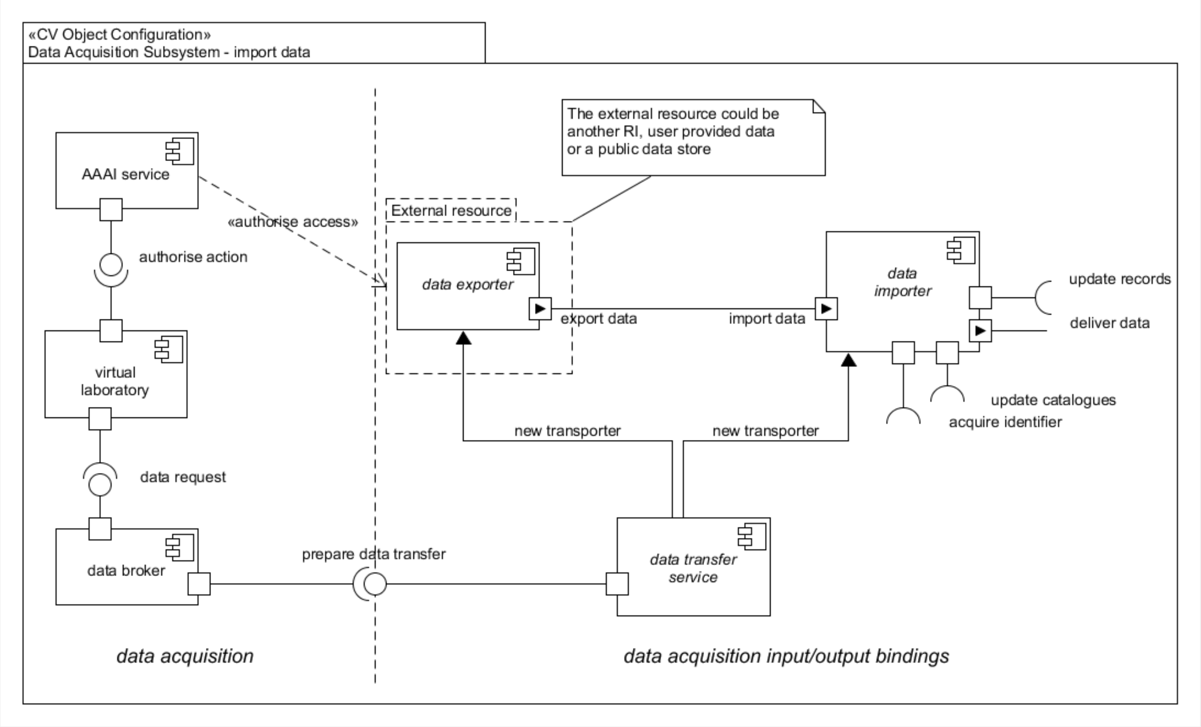
Data acquisition is computationally described as a set of **instrument controller** (encapsulating the accessible functionalities of instruments and other raw data sources out in the field), monitored and managed by one or more **acquisition service** (responsible for ensuring that any data is delivered into the infrastructure in accordance with current policies).

acquisition service invokes **data transfer services** which instantiate an appropriate **raw data collector** which retrieves data from the instrument controller. The four unlinked interfaces of the raw data collector will be linked to appropriate objects of the **CV Data Curation** subsystem.

Data acquisition from external resources

The diagram shows the organisation of six CV objects which are used for collecting data from an external resource. The external resource could be another RI, user uploaded data, or a public data store. The external resource could also be an interface for user observations provided by the RI, for instance for citizen observers

Data Acquisition Subsystem - import data



Notation

The six components used to model data acquisition from external resources. The external resource is a data source, not necessarily integrated into the infrastructure, providing data to data stores.

Acquisition is manipulated via a **virtual laboratory**, a community proxy, by which authorised agents can submit data to the RI. The **virtual laboratory** invokes a **AAAI service** to retrieve the appropriate credentials for accessing the external resource's **data exporter** and the internal **data importer**. After obtaining the credentials, the **virtual laboratory** invokes a **data broker** which in turn contacts a **data transfer service** which instantiates the appropriate **data exporter** and **data importer** objects and coordinates the transfer of data.

The four unlinked interfaces of the **data importer** will be linked to appropriate objects of the **Data Curation** subsystem.

CV Data Curation

One of the primary responsibilities of an environmental research infrastructure is the curation of the significant corpus of acquired data and derived results harvested from the data acquisition phase of the data lifecycle, data processing and community contributions. Scientific data must be collected, catalogued and made accessible to all authorised users. The accessibility requirement in particular dictates that infrastructures provide facilities to ensure easy availability of data, generally by replication (for optimised retrieval and failure-tolerance), publishing of persistent identifiers (to aid discovery) and cataloguing (aiding discovery and allowing more sophisticated requests to be made over the entirety of curated data). The following examples present two of the main functionalities of the data curation subsystem: data preservation and data annotation.

Data Preservation

The diagram shows the organisation of five CV objects which participate in the preservation of research data. The **data transporter** in the diagram could be replaced by a **raw data collector** or a **data importer** object, and the change would not affect the integrity of the system. Consequently, this configuration supports both types of data acquisition described in the data acquisition subsystem section. In the example there are no **Presentation Objects** which implies that the data preservation process is automated.

The **data transporter** modelled in the diagram is a complex device which at the same time invokes the **PID service**, the **catalogue service** , and the **data store controller**. The **PID service** is invoked to acquire a unique identifier for the incoming data set. The **catalogue service** i s invoked to store the metadata associated with the incoming data set. The **data store_controller** is invoked to store the incoming data set, along with its persistent identifier and liked to its associated metadata.

Data Curation Subsystem - data preservation

Notation

Data annotation

The diagram shows the organisation of five CV objects which participate in the annotation of research data. This task is carried with the oversight of a user or on request from a user, this is why the presentation object **sematic laboratory** is included. The **sematic laboratory** invokes a **semantic broker** which in turn invokes the **annotation service**. The **annotation service** provides two functionalities annotation and updating of the conceptual model, both the annotation and conceptual model are special types of metadata which are stored in the RI's catalogues and linked to a specific dataset, for this the annotation service invokes the **catalogue service** and the **data store controller**.

Data Curation Subsystem - data annotation

Notation

CV Data Publishing

Aside from the curation of scientific data, a research infrastructure must provide means to access that data. Access can be provided in a number of ways, including the export of curated datasets and the querying of data catalogues. Beyond the actual mechanism of access however are the issues of discovery and interpretation. Specific datasets may be found via citation (the publication of persistent identifiers associated with data) or by browsing data catalogues (permitting queries over multiple datasets). Additionally, a functionality to allow identifying the location of specific datasets in data stores should exist. It should also be possible to identify the ontologies, taxonomies and other semantic metadata associated with datasets or data requests and provide some form of mapping between representations as necessary.

The data publishing objects provide **CV Broker Objects** which mediate between data stores and catalogues and presentation objects (**virtual laboratories**). **data broker** act as intermediaries for access to data held within the data store objects supporting **CV Data Curation**. **Semantic brokers** enable semantic interpretation. Brokers are responsible for verifying the agents making access requests and for validating those requests prior to sending them on to the relevant data curation service.

The following examples present two important groups of functionalities provided by the data publishing subsystem: concept mapping and data publishing.

Concept Mapping

The **semantic laboratory** facilitates three actives which support linking data and metadata to one or more global models: (1) Build Global Conceptual Model, (2) Setup Mapping Rule, and (3) Perform Mapping. The **semantic broker** will facilitate updating the data and internal concept model to preserve the mappings by invoking the **catalogue service** and the **data store controller**.

Data Publishing Subsystem - concept mapping

Notation

Publishing Data and Metadata

The **virtual laboratory** facilitates the actives which support publishing data and metadata: (1) reviewing the data to be published, (2) publishing data, and (3) publishing metadata. The **data broker** will facilitate updating the data and metadata catalogues to stablish that the data has been reviewed, as well as storing the new links that make data and metadata publicly accessible. These actions are performed by invoking the appropriate **catalogue service** and the **data store controller**.

Data Publishing Subsystem - publishing data

Notation

CV Data Processing

The processing of data can be tightly integrated into data handling systems, or can be delegated to a separate set of services invoked on demand. In general, the more complicated processing tasks will require the use of separated services. The provision of dedicated processing services becomes significantly more important when large quantities of data are being curated within a research infrastructure. Scientific data is an example which is often subject to extensive post-processing and analysis in order to extract new results. The data processing objects of an infrastructure encapsulate the dedicated processing services made available to that infrastructure, either within the infrastructure itself or delegated to a client infrastructure.

Data Processing Objects

Notation

CV data processing objects are described as a set of **process controller** (representing the computational functionality of registered execution resources) monitored and managed by a **coordination service**. The coordination service delegates all processing tasks sent to particular execution resources, coordinates multi-stage workflows and initiates execution. Data may need to be **staged** onto individual execution resources and results **persisted** for future use; data channels can be established with resources via their process controllers. The following diagrams shows the staging and persistence of data.

Data Staging

The internal staging of data within an infrastructure for processing requires coordination between data processing components (which handle the actual processing workflow) and data curation components (which hold data within the infrastructure). The diagram bellow displays these two groups of objects which integrate part of the processing subsystem.

Data processing requests generally originate from **experiment laboratory** which validate requests by invoking an **AAAI service** . The **experiment laboratory** will send a process request to a **coordination service** , which interprets the request and starts a processing workflow by invoking the required **process controller**. Data will be retrieved from the data store and passed to the execution platform, the **coordination service** will request that a **data transfer service** to prepare a data transfer.

Data will be retrieved from the data store and passed to the execution platform, the **coordination service** will request that a **data transfer service** to prepare a data transfer. The **data transfer service** will then configure and deploy a **data exporter** which will handle the transfer of data between the storage and execution platforms, i.e. performing data staging. A data-flow is established between all required **data store controller**s and **process controller** via the **data exporter**. After the data-flow is established, processing starts. Processing can include a host of activities such as summarising, mining, charting, mapping, amongst many others. The details are left open to allow the modelling of any processing procedure. The expected output of the processing activities is a derived data product, which in turn will need to be persisted into the RIs data stores.

Data Processing Subsystem - data staging

Notation

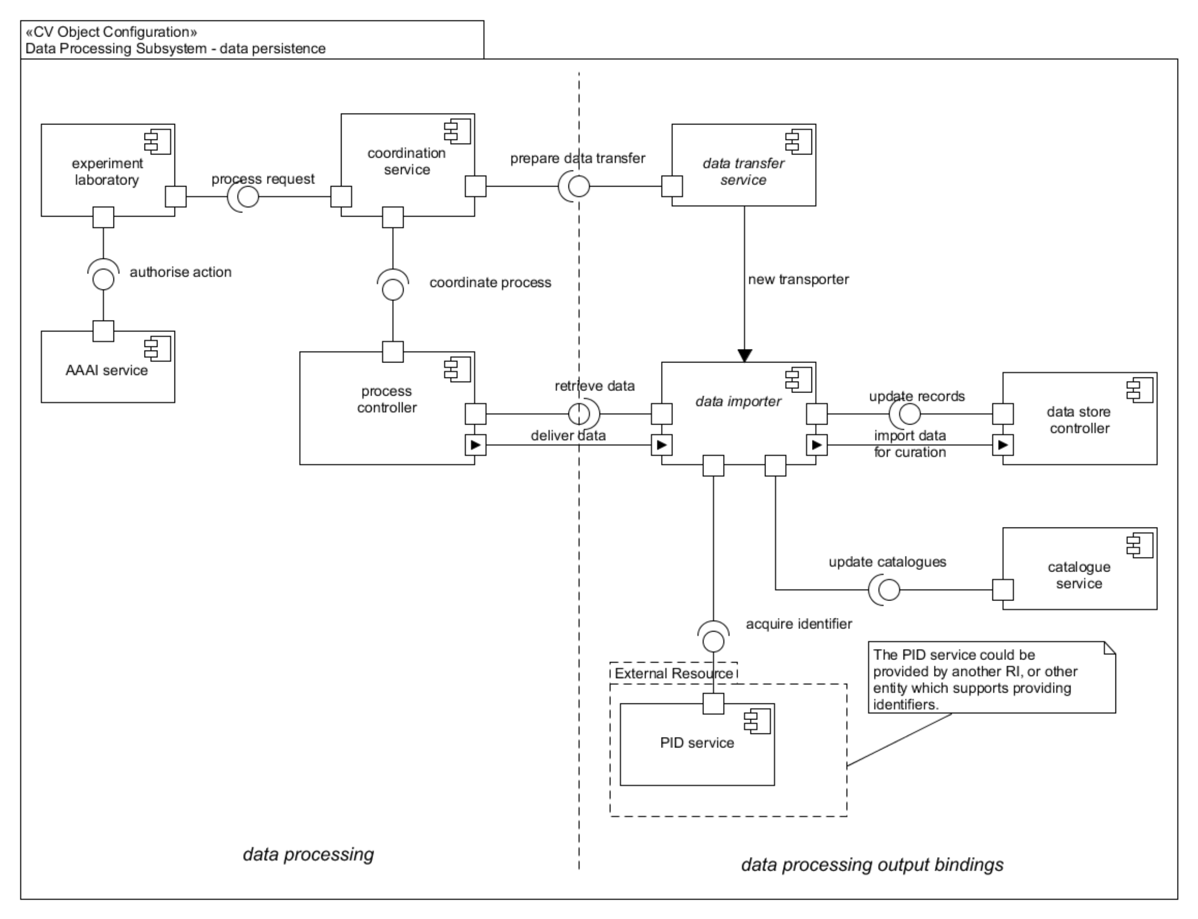
Data Persistence

The persistence of derived data products produced after processing of data within an infrastructure also requires coordination between data processing components (which handle the actual processing workflow) and data curation components (which hold data within the infrastructure). The diagram bellow displays these two groups of objects which integrate part of the processing subsystem.

Data processing requests generally originate from **experiment laboratory** which validate requests by invoking an **AAAI service** . The **experiment laboratory** can present results and ask the user if the results need to be stored, alternatively the user may configure the service to automatically store the resulting data. In either case, after processing, the **experiment laboratory** will send a process request to the **coordination service** , which interprets the request and invokes the **process controller** which will get the result data ready for transfer.

The **data transfer service** will then configure and deploy a **data importer** which will handle the transfer of data between the execution and storage platforms. A data-flow is established between **process controller** and **data store controller** via the **data importer**. After the data-flow is established, the data transfer starts. The persistence if data will trigger various curation activities including data storage, backup, updating of catalogues, requiring identifiers and updating records. These activities can occurs automatically or just as signals sent out to warn human users that an action is expected.

Data Processing Subsystem - data persistence



Notation

CV Data Use

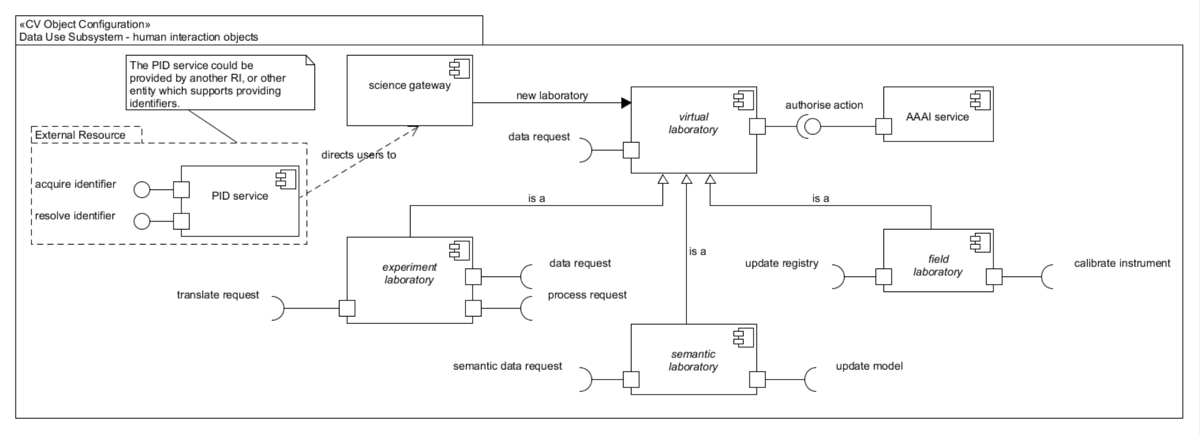
A research infrastructure is not an isolated entity, a research infrastructure aims to interact with the broader scientific community. In the ENVRI RM, a **science gateway** (Also known as *virtual research environment*) is assumed to be the main interaction platform for end users (in essence a scientific community portal). The **science gateway** is usually web-based and provides a number of services both for human users and for remote procedure invocation. These services may range from fundamental (data discovery and retrieval) to more interactive (user contribution and dataset annotation) to more 'social' (concerning user profiling, reputation mechanisms and workflow sharing).

The data use components are part of the **Presentation Object** and **Service Object** layers. The **Presentation Object Layer** includes different types of human interfaces aimed at providing access to the internal RI resources and services. The **CV Service Object Layer** encapsulates services provided for outside entities that require programmatic interaction with the RI.

In this sense, the data use subsystem can be subdivided in two object categories: **human interaction objects** and **service objects**.

Human Interaction Objects

Data Use Subsystem - human interaction objects



Notation

In the ENVRI RM, more complex interactions between the components facilitating data use and other components are mediated by **virtual laboratory**; these objects are deployed by **science gateway** in order to provide a persistent context for such interactions between certain groups of users and particular components within the RI. The Reference Model recognises the following specific sub-classes of laboratory:

- **field aboratory** (so-named because they interact with raw data sources 'in the field') are used to interact with the **Data Acquisition** components, allowing researchers to deploy, calibrate and un-deploy instruments as part of the integrated data acquisition network used by an infrastructure to collect its primary 'raw' data. Field laboratories have the ability to instantiate new **instrument controller** from the data acquisition set.
- **experiment laboratory** are used to interact both with curated data and data processing facilities, allowing researchers to deploy datasets for processing and acquire results from computational experimentation.
- **semantic laboratory** are used to interact with the semantic models used by a research infrastructure to interpret datasets and characteristic (meta)data.

Regardless of provenance, all laboratories must interact with an **AAAI ervice** in order to authorise requests and authenticate users of the laboratory before they can proceed with any privileged activities.

A **PID service** provides globally-readable persistent identifiers (PIDs) to infrastructure entities, mainly datasets, that may be cited by the community. PIDs can also be assigned to processes, services and data sources. This service is assumed to be provided by an external party, and is expected to direct agents attempting to read citations to one of the infrastructure's science gateways.

Service Objects

A constantly increasing portion of the interactions with an RIs are expected to be carried out by external systems interacting with data and other resources. In this case, the **Service Objects** becomes relevant, services are meant to provide access to external systems. In this case, external systems can include other RIs, universities, government agencies, industry applications, or other research groups which need to exploit the RIs data resources using client programs and the internet as a means to get to those data resources. In this form of integration, external systems are expected to implement **Presentation Objects** and **broker** objects which communicate with the RI services using public interfaces.

The following diagram shows an example of the use of service objects to connect an external system which will supply data to an RI. The components of the diagram are the same of those used internally for **acquisition from external source**, the difference is that the **virtual laboratory** , **data broker**, and **data exporter** objects are all part of an external system. These components interact with the **AAAI service** an d **data transfer service** services. The **AAAI service** will authorise the requested action and provide the required credentials. The **data transfer service** will establish the data interchange channel between the external **data exporter** and the internal **data importer** objects.

Data Use Subsystem - service objects example

Notation

CV Integration points

The CV defines the interfaces that support mutual invocation of CV objects functionality, allowing the composition objects to support complex interactions. Examination of these interfaces permits a set of possible bindings to be derived; for each of these bindings, the interaction between the bound objects can be specified in order to define the objects' behaviour when such a binding occurs. This then serves as a basis by which to synthesise the computational behaviour of the entire RI under different use-cases. The CV describes these use cases in detail by providing six integration models. These interactions can occur between lifecycle phases provided by a single RI, but also allow integration of components provided by third parties. The interactions define compound bindings between objects that allow the movement of scientific dataset between different parts of a research infrastructure.

- **CV Brokered Data Export** (the export of user-requested data)
- **CV Brokered Data Import** (the import of user-provided data)
- **CV Brokered Data Query** (the querying of curated data by users)
- **CV Citation (the resolution** of data and resources cited in publications)
- **CV Instrument Integration** (the integration of new instruments for data acquisition into the infrastructure)
- **CV Raw Data Collection** (the acquisition of raw data from integrated data sources)

The aggregation of these core interactions form a minimal computational model for environmental science research infrastructures that can be used as a starting point for modelling real infrastructures.

CV Brokered Data Export

Exporting data out of a research infrastructure entails retrieving data from the data curation subsystem and delivering it to an external resource. This process must be brokered by the data use and data publishing subsystems.

Brokered Data Export

Notation

Generally requests for data to be exported to an external resource originate from a **virtual laboratory**. All requests are validated by the **AAAI service** via its *authorise action* interface. The laboratory provides an interface to an external resource (this might take the form of a URI and a preferred data transfer protocol) and submits a request to a data broker in the data publishing subsystem via its *data request* interface. The data broker will translate any valid requests into actions; in this scenario, a data transfer request is sent to the **data transfer service** within the data curation subsystem.

The data transfer service will configure and deploy a **data exporter**; this exporter will *retrieve data* from all necessary data stores, opening a data-flow from data store to external resource. The exporter is also responsible for the repackaging of exported datasets where necessary – this includes the integration of any additional metadata or provenance information stored separately within the infrastructure that needs to be packaged with a dataset if it is to be used independently of the infrastructure. As such, the exporter can invoke the **catalogue service** to retrieve additional meta-information via its *export metadata* interface.

CV Brokered Data Import

Importing data from sources other than the acquisition network requires that the import be brokered by the publishing subsystem before data can be delivered into the data curation subsystem.

Brokered Data Import

Notation

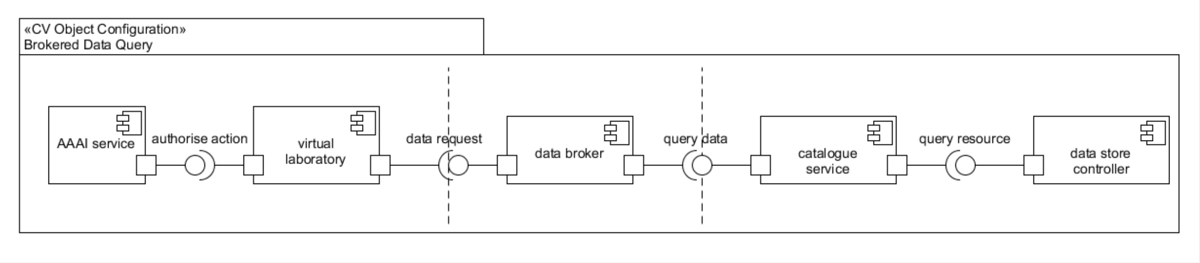
A **virtual laboratory** allows researchers to upload new data into a research infrastructure. All requests are validated by the **AAAI service** via its *authorise action* interface. The laboratory provides an interface to an external resource (this might take the form of a URI and a preferred data transfer protocol) and submits a request to a **data broker** in the data publishing subsystem via its *data request* interface. The data broker will translate any valid requests into actions; in this scenario, a data transfer request is sent to the **data transfer service** within the data curation subsystem.

The data transfer service will configure and deploy a **data importer,** the importer will open a data-flow from an external resource to one or more suitable data stores within the infrastructure and *update records* within those stores as appropriate. The importer is responsible for the annotation and registration of imported datasets – this generally entails obtaining a global persistent identifier for any new datasets and updating the catalogues used by the research infrastructure to identify and sort its data inventory. As such, the importer can invoke the **catalogue service** to *update catalogues* and invoke any community-used **PID service** to *acquire identifiers*.

CV Brokered Data Query

Querying curated data resources requires that the request be brokered by the data publishing subsystem before any results will be retrieved from the data curation subsystem and delivered to the client from which the source came.

Brokered Data Query



Notation

Virtual laboratory is able to query the data held within a research infrastructure subject to access privileges governed by the **AAAIservice** (invoked via its *authorise action* interface). Data requests are forwarded to a **data broker** within the data publishing subsystem, which will interpret the request and contact any internal services needed to fulfil it. In this case, the data broker will invoke the **catalogue service** via its *query data* interface; the catalogue service will locate the datasets needed to answer any given query and then proceed to *query resources* within infrastructure **data store controller**.

CV Instrument Integration

CV Data Acquisition relies on an integrated network of data sources (referred to generically as 'instruments') that provide raw measurements and observations continuously or on demand. This network is not necessarily static; new instruments can be deployed and existing instruments can be taken off-line or re-calibrated throughout the lifespan of a research infrastructure. In the Reference Model, modifications to the acquisition network should be performed via a 'virtual laboratory' that permits authorised agents to oversee acquisition and calibrate instruments based on current community practice or environmental conditions.

Instrument Integration

Notation

Instruments can be added to and removed from a data acquisition network by a **field laboratory** accessed via a **science gateway**. The field laboratory must be able to provide an **instrument controller** for any new instrument added in order to allow the data acquisition subsystem to interact with the instrument. Deployment, un-deployment or re-calibration of instruments requires authorisation - this can only be provided a valid **AAAI service** (via its *authorise action* interface). Any changes to the data acquisition network must be registered with an **acquisition service** (via its *update registry* interface).

The behaviour of an instrument controller can be configured by the acquisition service by invoking functions on the controller via its *configure controller* interface.

A field laboratory also provides the means to calibrate instruments based on scientific best practice where applicable - this is done via the instrument controller's *calibrate instrument* interface.

CV Citation

The citation of datasets involves reference to persistent identifiers assigned to objects within a research infrastructure. Such citations are resolved by referring back to the infrastructure, which can then return a report describing the data cited.

Data Citation

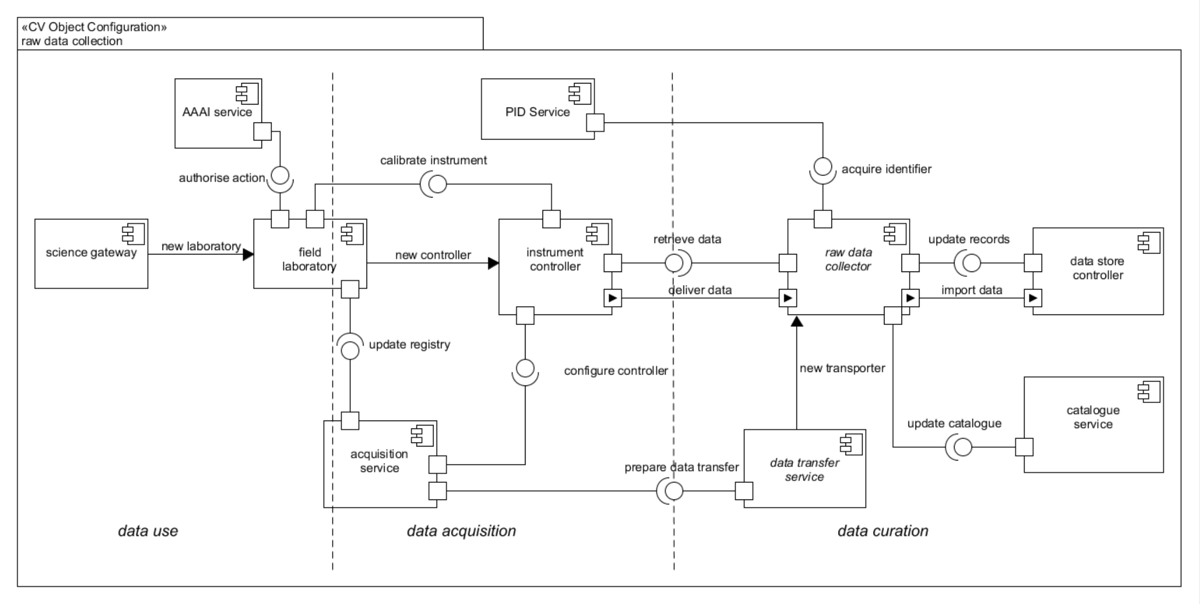
Notation

A user or external service tries to *resolve an identifier* (found in a citation) with the global **PID service** used by the research infrastructure. By dereferencing the given identifier, that user or service is directed to a **science gateway** used to interact with the infrastructure. From there, the desired provenance information about the citation can be immediately retrieved, or a **virtual laboratory** can be deployed for more complex interactions with the research infrastructure.

CV Raw Data Collection

The collection of raw scientific data requires coordination between the CV Data Acquisition phase (which extracts the raw data from instruments) and the CV Data Curation phase (which packages and stores the data).

Raw Data Collection



Notation

The delivery of raw data into a research infrastructure is driven by collaboration between an **acquisition service** and a **data transfer service**. This process can be configured using a **field laboratory** subject to an **AAAI service** authorisation, via the **AAAI service**'s *authorise action* interface. Regardless, the acquisition service identifies the instruments that act as data sources and provides information on their output behaviour, whilst the data transfer service provides a **data transporter** that can establish (multiple, persistent) data channels between instruments and data stores. The data transporter (a **raw data collector**) can initiate data transfer by requesting data from one or more **instrument controller** and preparing one or more **data store controller** to receive the data.

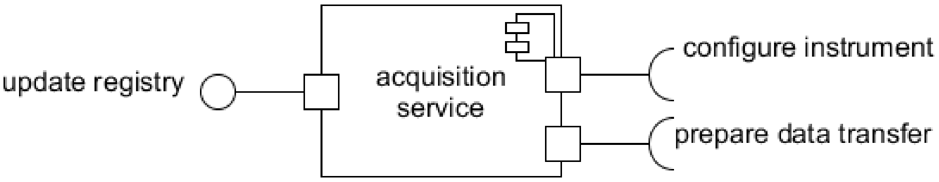
The raw data collector is considered responsible for packaging any raw data obtained into a format suitable for curation - this may entail chunking data streams, assigning persistent identifiers and associating metadata to the resulting datasets. To assist in this, a raw data collector may acquire identifiers from a **PID service**. It may also want to register the presence of new data and any immediately apparent data characteristics in infrastructure data catalogues - this is done by invoking an update operation on the **catalogue service**.

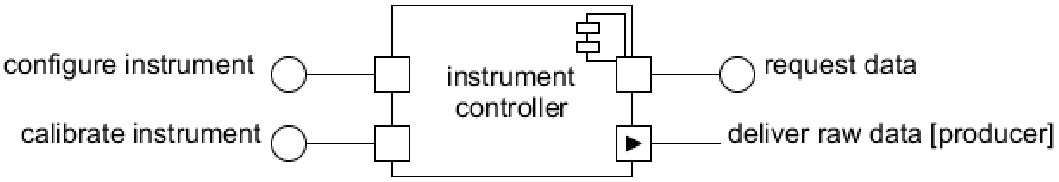
How to read the Model (Computational Viewpoint)

The computational viewpoint (CV) is concerned with the modelling of computational objects and the interactions between their interfaces, according to the ODP specification [37]. The ENVRI RM uses a lightweight subset of the full ODP specification to model the abstract computational requirements of an archetypical environmental science research infrastructure.

> The encapsulation of computational objects (and interfaces) occurs at a conceptual level rather than the implementation level – it is perfectly admissible for the functions of a given object to be distributed across multiple computational resources in an implemented infrastructure, should that be supported by its architecture, if that distribution does not interfere with the ability to implement all of that object's interfaces (and thus behaviours). Likewise the functionalities of multiple objects can be gathered within a single implemented service, should that be desired.

The first-class entity of the CV is the *computational object*:





> In diagrams, each a computational object is represented using a rectangle with a decoration on the upper right corner.. The text within the object indicates the name of the object. The decoration on the upper right corner is standard UML notation for component.

A computational object encapsulates a set of functions that need to be collectively implemented by a service or resource within an infrastructure. To access these functions, a computational object also provides a number of *operational* interfaces by which that functionality can be invoked; the object also provides a number of operational interfaces by which it can itself invoke functions on other objects. Each computational object may also have *stream* interfaces for ferrying large volumes of data within the infrastructure. In summary:

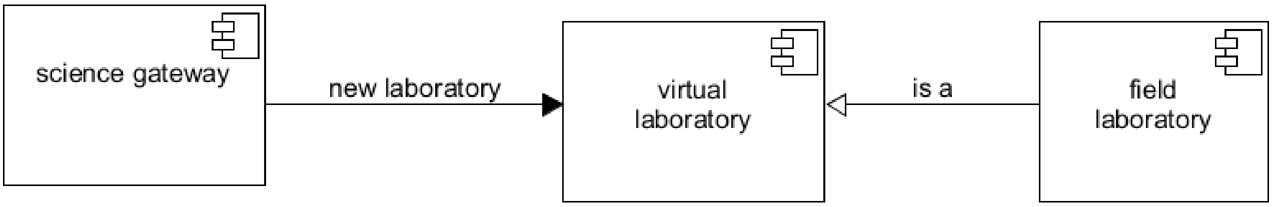
- **Operational interfaces** are used to pass messages between objects used to coordinate general infrastructure operations such as querying a data resource or configuring a service. A given operation interface must be either a *server* interface (providing access to functions that can be invoked by other objects) or a *client* interface (providing a means by which an object operations can be invoked on other objects).

  > In diagrams, client and server interfaces are linked using 'ball and socket' notation: clients expose sockets (half-circles) whilst servers expose balls (complete circles).

- **Stream interfaces** are used to deliver datasets from one part of the infrastructure to another. A *producer* interface streams data to one or more bound *consumer* interfaces as long as there is data to transfer and all required consumers are available to receive that data (whether one, all or some of the consumers must be available depends on the circumstances of the data transfer). Data channels are typically established by operations invoked via operational interfaces (which typically negotiate the terms of the transfer), but can persist independently of them (which is useful for long-term continuous transfers such as from sensor networks to data stores).

  > In diagrams, producer and consumer stream interfaces are linked using a double-arrow notation: the arrow-head points away from producers, towards consumers.
  >
  > The decoration on the port boxes is not standard UML but is used to distinguish streaming interfaces.

As well as having interfaces by which to interact with other objects, some computational objects possess the right to create other computational objects; this is done typically to deploy transitory services or to demonstrate how an infrastructure might extend its functionality.



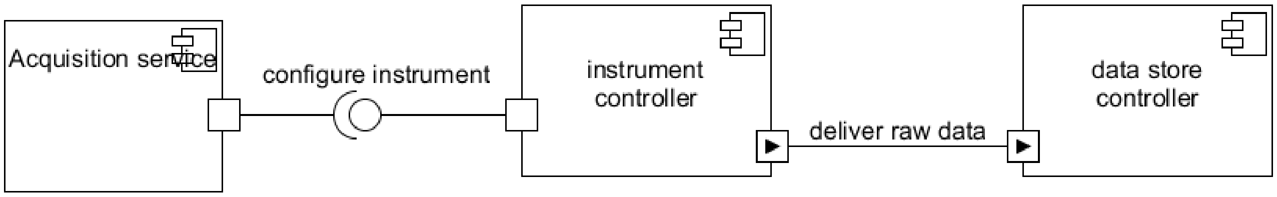
Some objects extend the functionality of other objects; these objects possess all the interfaces of the parent (usually in addition to some of their own) and can be created by the same source object if the capability exists.

> In diagrams, the ability to create objects is noted by a single filled arrow extending from the creating object to the object being created, with the annotation 'new <object>'. If one object extends another, then this can be illustrated using an unfilled arrow from the sub-object to the parent, with the annotation 'is a'.

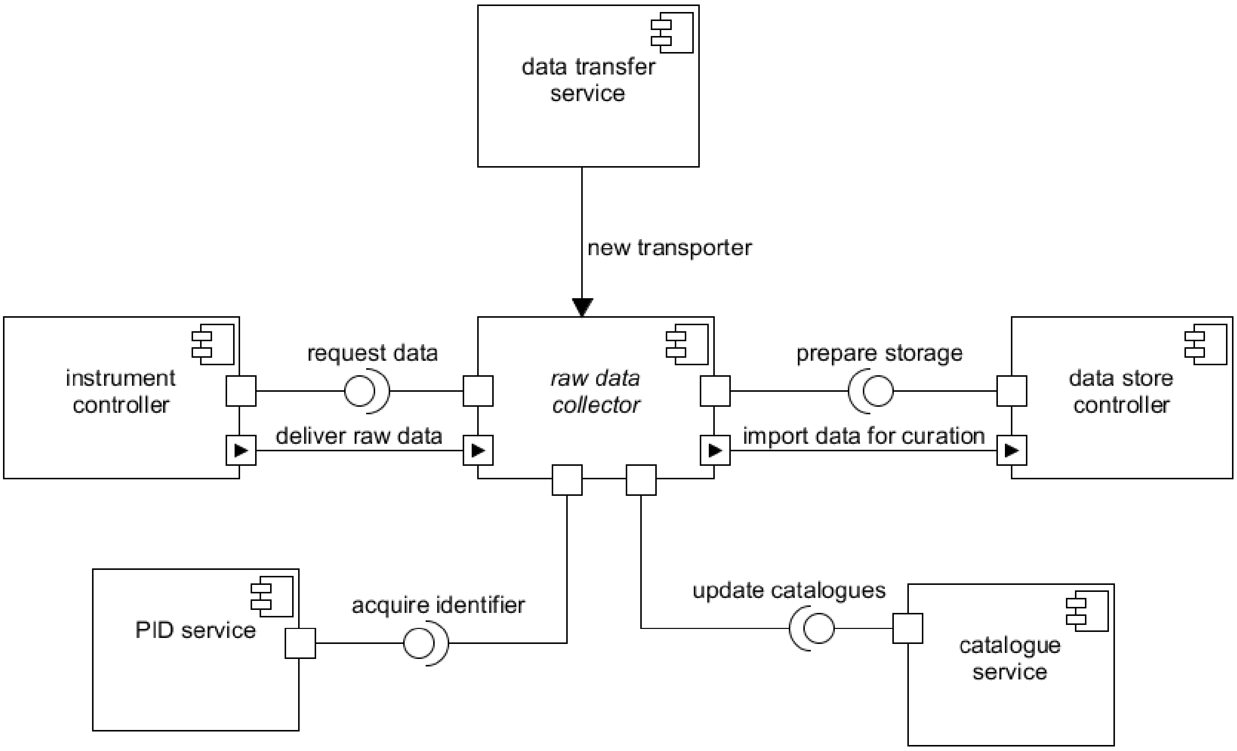
Each interface on a computational object supports a certain type of interaction between objects, which determine the bindings that can be made between interfaces. A *binding* is simply an established connection between two or more interfaces in order to support a specific interaction between two or more computational objects. A client operational interface can be bound to any server operational interface that provides access to the functions that the client requires. Likewise a producer stream interface can be bound to any consumer stream interface that can consume the data produced by the former.

> For simplicity, client and server interfaces designed to work together in the Model share the same name; thus a a client interface ***x*** can bind to any server interface ***x*** and a producer interface ***y*** can bind to any consumer interface ***y***. When a binding is explicitly shown in a diagram, the binding itself is identified by that shared name.

Once bound via their corresponding interfaces, two objects can invoke functions on one another to achieve some task (such as configuration of an instrument or establishment of a persistent data movement channel).



Primitive bindings can be established between any client/server pair or producer/consumer pair as appropriate. Compound bindings between three or more interfaces can be realised via the creation of *binding objects*, a special class of transitory computational object that can be used to coordinate complex interactions by providing primitive bindings to all required interfaces.



The use of binding objects removes the imperative to decompose complex interactions into sets of pairwise bindings between objects; this suits the level of abstraction at which the Model is targeted, given that the specific distribution of control between interacting objects is often idiosyncratic to different infrastructure architectures.

> The names of binding objects are typically *italicised* in diagrams to better distinguish them from 'basic' computational objects.

A note about implementation

In principle, all computational objects and their interfaces can be implemented as services or agents within a service-oriented architecture –

this is not required however. Certain objects may be implemented by working groups or even individuals within the infrastructure organisation, bindings between their interfaces implemented by physical interactions, or otherwise human-oriented processes (such as sending data via email).

For example, in the Model, a *field laboratory* has the ability to calibrate instruments (represented by *instrument controllers*) via a binding of their common *calibrate instrument* interfaces. Potentially, the field laboratory could be implemented by a virtual research environment within which authorised users can interact online with instruments deployed in the field, modifying how they acquire data. In practice, the `field laboratory' may simply abstractly represent the activities of field agents (scientists and technicians) who actually travel to sites where instruments are deployed and manually make adjustments.

This possibility of this kind of 'human-driven' implementation of interactions between computational objects should be accounted for when considering the 'computational' viewpoint of a research infrastructure.

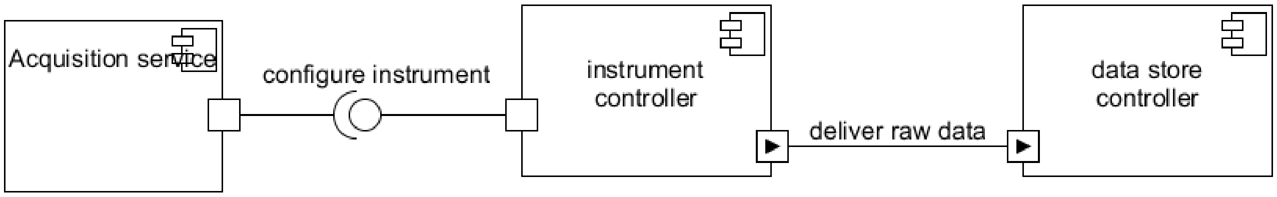
How to use the Model (Computational Viewpoint)

The computational viewpoint of the Model identifies a standard set of components and interfaces from which can be derived a standard set of interactions that a research infrastructure design should address. The Model does *not* specify how those interactions should be implemented – indeed, over the course of the lifetime of a research infrastructure, implementations may change. Nevertheless, the set of the most important interactions should remain constant regardless of implementation changes.

Someone trying to apply the Computational Viewpoint of the Model to their existing or planned research infrastructure should conduct two primary activities: mapping agents and services to computational objects, and defining the interactions that should occur when two or more interfaces are bound together.

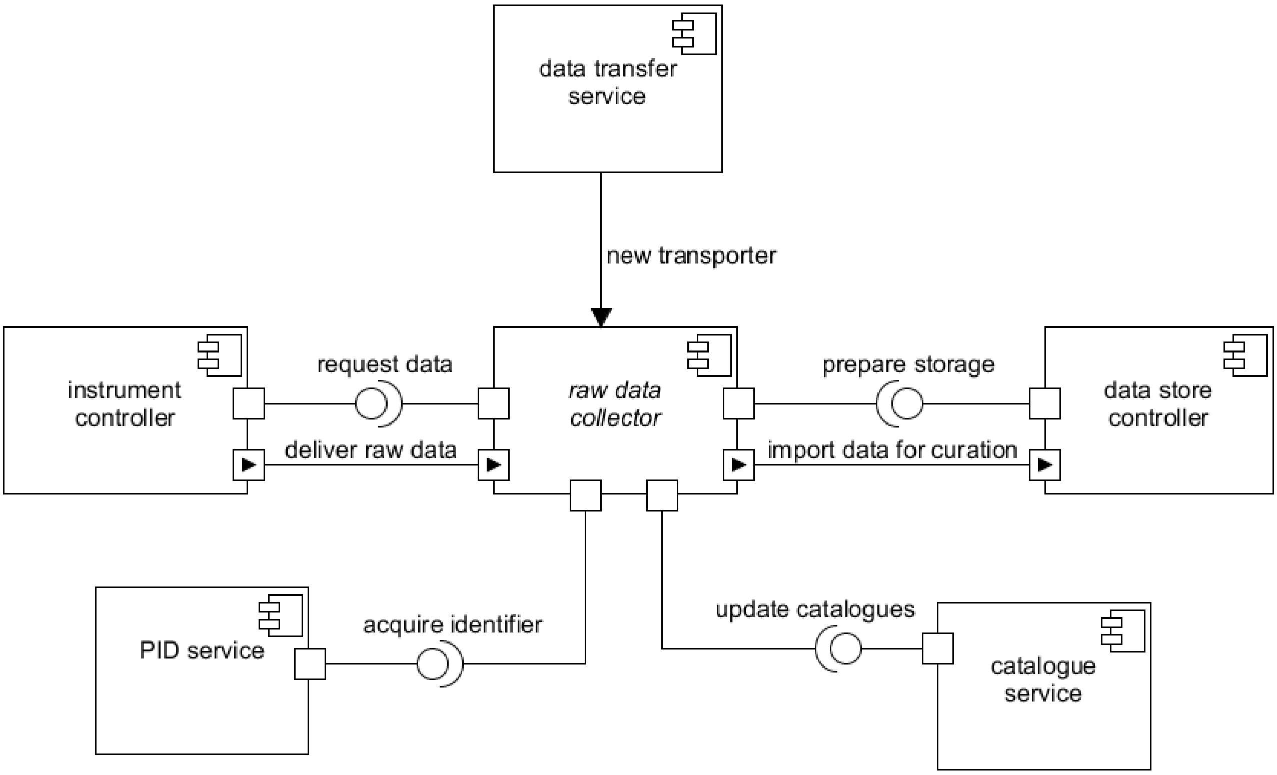
For each computational object in the Model, there should be at least one component or service (or group thereof) provided by the infrastructure that can provide the functions described – depending on the architecture of the infrastructure, there may be multiple candidate, particularly for federated infrastructures. Every such candidate could provide an instantiation of the given object. If no candidates exist, then either (a) the infrastructure does not provide the service embodied by the computational object (and it should be clearly understood that this is indeed the case) or (b) the infrastructure is missing functionality that should be implemented to bring it in compliance with the Model.

For each compatible pair of interfaces (operational or stream), there exists an interaction that should occur given a binding between those two interfaces. The Model does *not* prescribe these interactions, instead simply providing the means to identify them. A compliant research infrastructure should in principle have a well-defined description for *every* possible binding between interfaces on objects that it provides an implementation for.



In the above diagram, an (operational) primitive binding has been established between the *configure instrument* interfaces of an *acquisition service* object and an *instrument controller* object, as well as a (stream) primitive binding between the *deliver raw data* interfaces of the *acqui sition service* and a *data store controller* (see How to read the Model (Computational Viewpoint) to understand the above notation and terms). Thus, assuming a Model-compliant research infrastructure that provides at least one acquisition service and instrument controller, there should be a specification of what happens when a 'configure instrument' binding occurs between an acquisition service and instrument controller. Likewise, there should be a specification of how raw data is delivered from an instrument (represented by the instrument controller) to a data store (represented by its own controller).

Many *primitive* (two-interface) bindings are linked in that the establishment of one binding will necessarily lead to the establishment of other bindings, implying a unified interaction description. This is particularly true for *compound* bindings where a particular binding object is created to establish pairwise primitive bindings with multiple computational objects that must all contribute to the given interaction. A compliant research infrastructure must therefore identify all such compound bindings and should define how any binding objects created to coordinate interactions are instantiated (generally as either an oversight service or as 'abstractly' as a distributed process involving agents / services participating in the resulting interaction.



In the above diagram, there exist multiple primitive bindings to a central binding object (the *raw data collector*) that nonetheless all relate to a single compound interaction (describing how the transfer of data from an instrument to a data store is configured and managed). It is very important to properly describe the relationship between the individual bindings and how the compound interaction between the various computational objects involved is produced if constructions like in the diagram above are to be properly understood. In the reference material for the Model, a number of 'core' reference interactions have been described informally to provide a Computational Viewpoint for Model implementors.

Interaction specifications (whether for primitive or compound interface bindings) can take any form deemed suitable by the developers of the infrastructure – for example, UML diagrams such as activity or sequence diagrams may be appropriate, as might be a formal logic model or BPEL workflow, or even natural language if the interaction is simple enough.

Engineering Viewpoint (Draft)

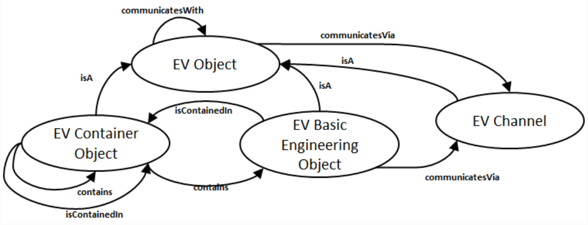
The Engineering Viewpoint is concerned with transparently distributing the objects defined in the computational viewpoint (**C omputational Viewpoint**) over nodes, either physical and/or virtual. It is concerned with the structure of the nodes. The definition of the engineering viewpoint includes the description of structures needed to supporting information viewpoint objects objects (**IV objects**), and for the specification of the communicating channels between nodes. The Engineering Viewpoint is also concerned with supporting non-functional requirements relating to performance, reliability, load-balancing, and other similar, which correspond to the objectives and policies defined in the **SV Communities** [01]

In specifying the Engineering viewpoint (EV) in the ENVRI RM, the main aim is to control and constrain the architectural style to be adopted at interaction points between RIs to improve the interoperability between RIs; without the need of investment in expensive peer-to-peer transformation/interworking/gateway functionalities. The aim is NOT to control the internal engineering of individual RIs, and thus less emphasis will be given to these aspects when specifying the EV.

This engineering viewpoint specification is designed to research data management functionalities which are commonly shared among more than one RI (Identification and Citation, Curation, Cataloguing, Processing, and Provenance).

The presentation of the Engineering Viewpoint is divided in three parts: container structure, engineering objects, and object configuration.

The engineering viewpoint defines four types of objects: engineering object (EV object) basic engineering object (EV BEO) container object (EV container) and channel object (EV Channel). EV objects enable distribution among container objects, they provide functions that control the deployment of basic engineering objects such as instantiating, starting, pausing, restarting and stopping services. Basic engineering objects map one to one to Computational Objects and they are designed to implement the functionalities described in the Computational Viewpoint. Container objects provide an organisational structure that can be used to distribute the computational functionalities (provided by BEO). Container objects can be further classified as Node, Capsule and Cluster. Finally, channel objects define the communication structures to pass messages between node objects.



Engineering Viewpoint components and their relationships

- **Container Structure** provides the detailed description of the recommended of structure to be used for distributing the engineering objects .

- **Engineering Objects** describes the main objects to be modelled in the EV.
- **Object Configuration** illustrates how the services are used for supporting a set of functionalities which are commonly exposed and used by RIs

Note
Before proceeding, the reader may wish to study the pages on **RM ODP Background** that describe the theory used to support the definition of the engineering viewpoint.

[01] For an introduction to the Engineering and other viewpoints of ODP, refer to Linington et.al. [**37**].

Container Structure (Draft)

The first element to describe in the engineering viewpoint is the structure of the **Container Objects** that help structuring the systems supporting the RI processes.

An **Container Objects** is an structural engineering object. **Container Objects** help structuring the systems supporting the RI processes. There are four possible types of container objects: node, nucleus, capsule and cluster [01].

The ENVRI RM engineering viewpoint recommendation defines the distribution of objects as services. In this configuration, EV objects are distributed in four main containers which group customer integration interface objects, collaboration integration interface objects, management components and domain function components.

- **Customer Integration**: provides interface components (APIs) which allow the use of the functionality by different service clients. This can include exposing management functionalities so that the service can be controlled remotely.
- **Downstream Integration (Collaboration Integration)**: provided to allow the service to call other services and back end components. This includes calling persistence layer components, such as storage manager components.
- **Service Management**: provided for management and monitoring of the service. This allows controlling the service (start, stop, pause, restart) as well as providing data for troubleshooting, logging, and provenance.
- **Service Behaviour**: provided to implement service behaviour. This is where the main functionalities of the service are located.

Recommended container structure for functionalities exposed as services.

The specific type of each container is left open to allow greater flexibility in distribution at implementation time.

The proposed architectural container structure for the EV is based on the Microservice Architecture [02, 03]. However, the EV does not propose a full migration to Microservices architectures but a gradual approach in which functionalities are decoupled and provided as services in a piecemeal fashion. For instance, functionalities which are always dependent on external services such as data identification and citation, or common functionalities which can be shared among various RIs, such as processing, cataloguing, provenance or or AAAI.

[01] See Linington et.al. p. 94-95 [**37**]

[02] Sam Newman. (2012). Building Microservices, Designing Fine Grained Systems. O'Reilly, ISBN-10: 1491950358

[03] Neal Ford Building Microservice Architectures. ThoughtWorks. Voxxed days Vienna 2016. Video https://www.youtube.com/watch?v=pjN7CaGPFB4, slides: http://nealford.com/downloads/Building_Microservice_Architectures_Neal_Ford.pdf

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) [01, 02].

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**.

Engineering object groups

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 functionalites: 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	**data annotation** (behaviour)
Information	**metadata catalogue** (instance of object)
Computational	**annotation service**

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 catalougues, update catalogues, export metadata, and query resource.

Catalogue service functions

Function	Description	Type

Catalogue service correspondences

Viewpoint	Corresponding Object
Science	**catalogue system** (role)
Information	**metadata catalogue**(object)
Computational	**catalogue service**

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

Catalogue service correspondences

Viewpoint	Corresponding Object
Science	**data processing subsystem** (role) **coordinate service** (behaviour)
Information	**process data** (action)
Computational	**coordination service**

create process controller	function for creating and configuring the required **process controllers** 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 **data transporter**).	client
update records	function provided to inform downstream resources about impending asset transfers (inherited from **data transporter**).	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 exporter correspondences

Viewpoint	Corresponding Object
Science	**data publishing subsystem** (role) **data use subsystem** (role)
Computational	**data exporter**

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

data importer correspondences

Viewpoint	Corresponding Object
Science	**data acquisition subsystem** (role) **data curation subsystem** (role)

retrieve asset	function provided to enable requesting data from a designated data store (inherited from **data transporter**).	client
update records	function provided to inform downstream resources about impending asset transfers (inherited from **data transporter**).	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.	cllient
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

Computational	**data importer**

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 resouce, 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

data store controller correspondences

Viewpoint	Corresponding Object
Science	**data curation subsystem** (role)
Computational	**data store controller**

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 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 (**raw data collector**, **data importer**, or **data exporter**)	internal

data transfer service correspondences

Viewpoint	Corresponding Object
Science	**data curation subsystem**(role)
Computational	**data transfer service**

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
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

data transfer service correspondences

Viewpoint	Corresponding Object
Science	**PID Manager** (role)
Information	**assign unique identifier** (action)
Computational	**PID manager** (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

Catalogue service correspondences

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

Viewpoint	Corresponding Object
Science	**data processing subsystem** (role) **compose service**(behaviour)
Information	**process data** (action)
Computational	**process controller**

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 catalougues, 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

Catalogue service correspondences

Viewpoint	Corresponding Object
Science	**catalogue system** (role)
Information	**track provenance** (action) **metadata catalogue** (instance of object)
Computational	**catalogue_service** (instance of object)

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

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 **data transporter**).	client
update records	function provided to inform downstream resources about impending asset transfers (inherited from **data transporter**).	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.	cllient
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

raw data collector correspondences

Viewpoint	Corresponding Object
Science	**data acquisition subsystem** (role) **data curation subsystem** (role)
Computational	**raw data collector**

[01] 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

[02] 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

Object Configuration (Draft)

Object Configuration illustrates how the engineering objects are used for supporting the functionalities commonly shared among more than one RI such as Identification and Citation, Curation, **Cataloguing Configurations**, Processing, and Provenance.

Cataloguing Configurations

There are seven cataloguing processes (D8.3) which need to be supported using EV objects. These basic EV configurations are mapped to these processes in **Table 01**. As shown in the table, the combination of these basic configurations allows supporting different cataloguing processes.

Table 1. Basic Cataloguing Processes

Processes (User Stories/Requirements) (Stage 1 Description)	Configuration Name
1. Store an asset (e.g. dataset) with metadata sufficient for cataloging purposes;	**Store Asset Configuration**
2. Discover an asset using the metadata – the richer the metadata and the more elaborate the query the greater the precision in discovering the required asset(s);	**Discover Asset Configuration**
3. Download an asset based on metadata	**Download Asset Configuration**
4. Select/project an asset in situ location accessed via metadata	**Select Asset Configuration**
5. Move (selected parts of) asset to location accessed via metadata	**Move Asset Configuration**
6. Compose (selected parts of) asset into workflow accessed via metadata	**Compose Asset Configuration**
7. Update an asset and/or metadata	**Update Asset Configuration**

This section define set of seven independent cataloguing services that support each process on **Table 01**. This clearly separates the functionalities vertically and shows how the BEOs interact with each other. The structure of the following configurations is based on the **Container Structure**.

In addtion to the EV BEOs all the configurations require a management component object and a catalogue system object.These engineering objects together with the collection of configurations conform the complete catalogue service.

Store Asset Configuration

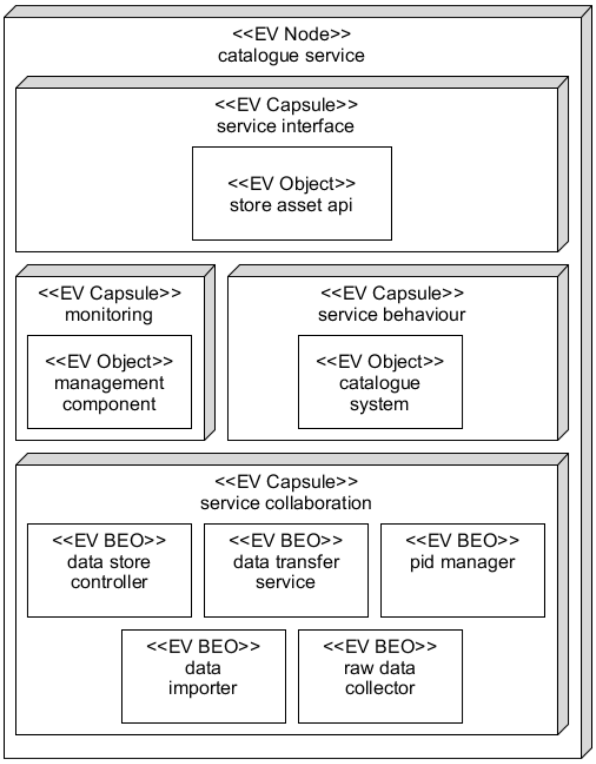
The configuration for storing a new asset depends on six BEOs: **catalogue service**, **data transfer service**, **raw data collector**, **data importer**, **pid manager**, and **data store controller**.

The figures on the right shows the configuration of the service using the recommended **Container Structure**.

This configuration shows a self contained service which already contains the required objects for three possible service variations: collect from sensor, import from another repository, and assign global unique pid.

Name:	Store Asset
Process:	Store an asset (e.g. dataset) with metadata sufficient for cataloging purposes

Resources	• asset • metadata
Outputs	• asset [registered] • metadata [registered] • asset pid [registered]
Options	• collect data from sensors • import data from external source • assign global unique pid



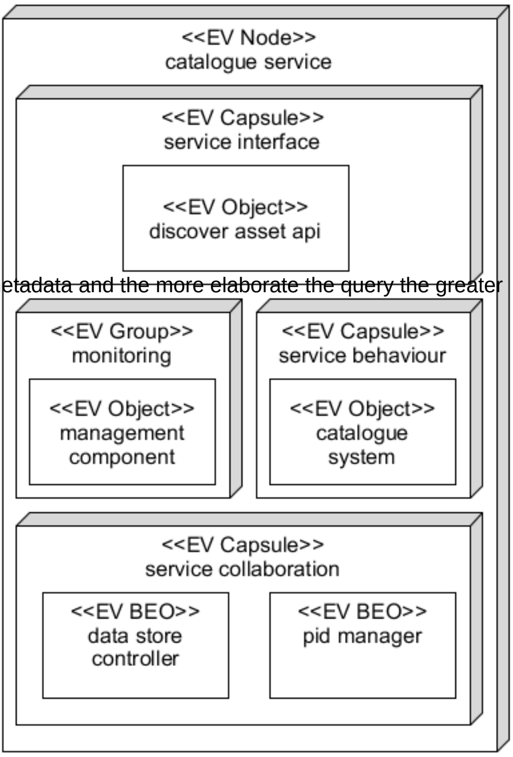
Discover Asset Configuration

The configuration for discovering an asset depends on three BEOs: **catalogue service**, **pid manager**, and **data store controller**.

The figures on the right shows the configuration of the service using the recommended **Container Structure**.

This configuration shows a self contained service which already contains the required objects for supporting the discovery process.

Name:	Discover Asset
Process:	Discover an asset using the metadata – the richer the metadata and the more elaborate the query the greater the precision in discovering the required asset(s)
Resources:	• query parameters (metadata)
Outputs:	• metadata • asset pid • asset reference (publishable) • asset description
Options:	• faceted discovery (filtering of results)
Correspondences	Technology Viewpoint: **Catalogue Discovery Object**

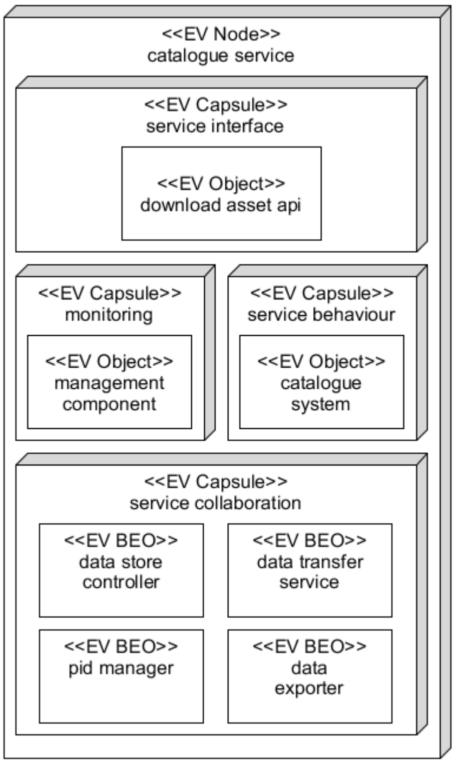


Download Asset Configuration

The configuration for storing a new asset depends on five BEOs: **catalogue service**, **data transfer service**, **data exporter**, **pid manager**, and **data store controller**

The figures on the right shows the configuration of the service using the recommended **Container Structure**.

This configuration shows a self contained service which already contains the required objects for supporting the discovery process.



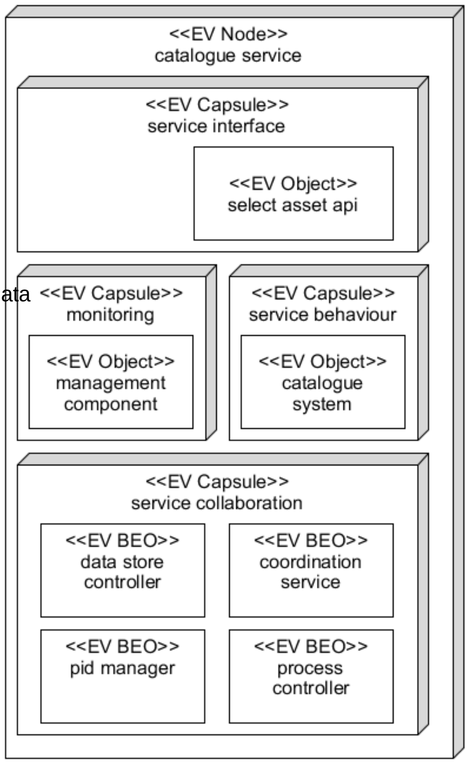
Name:	Download Asset
Process:	Download an asset based on metadata
Resources	• asset pid (precondition: discovery) • filtering parameters (metadata, optional)
Outputs	• asset (copy) • metadata • asset pid • asset reference (publishable) • asset description
Options	• download complete • download partial

Select Asset Configuration

The configuration for selecting an asset depends on five BEOs: **catalogue service**, **coordination service** , **process controller**, **pid manager**, and **data store controller**

The figures on the right shows the configuration of the service using the recommended **Container Structure**.

This configuration shows a self contained service which already contains the required objects for supporting the discovery process.

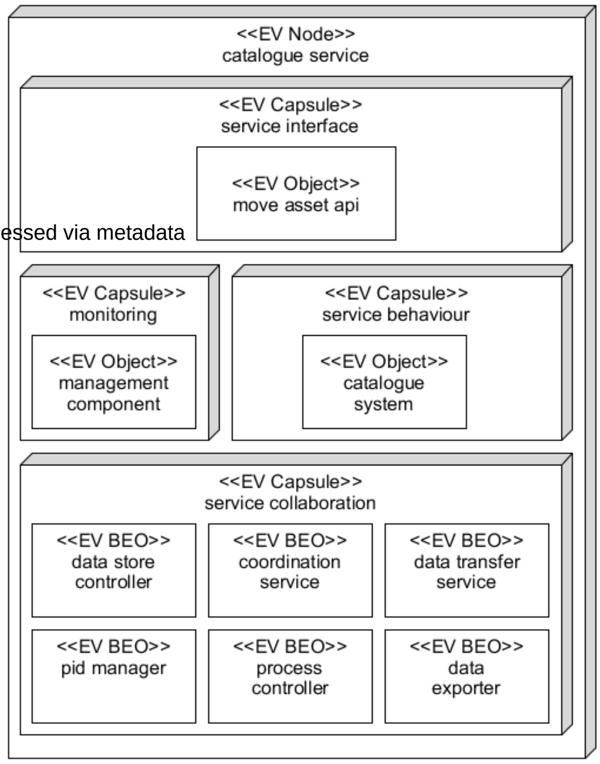


Name:	Select Asset
Process:	Select/project an asset in situ location accessed via metadata
Resources	• asset pid (precondition: discovery) • filtering parameters (metadata, optional)
Outputs	• asset (processed view) • metadata • asset pid • asset reference (publishable) • asset description • asset view details
Options	• select • project

Move Asset Configuration

The configuration for moving an asset depends on seven BEOs: **catalogue service**, **data transfer service** , **data exporter**, **pid manager**, **data store controller**, **coordination service**, and **pro cess controller**, The figures on the right shows the configuration of the service using the recommended **Container Structure**.

This configuration shows a self contained service which already contains the required objects for supporting the move asset process.



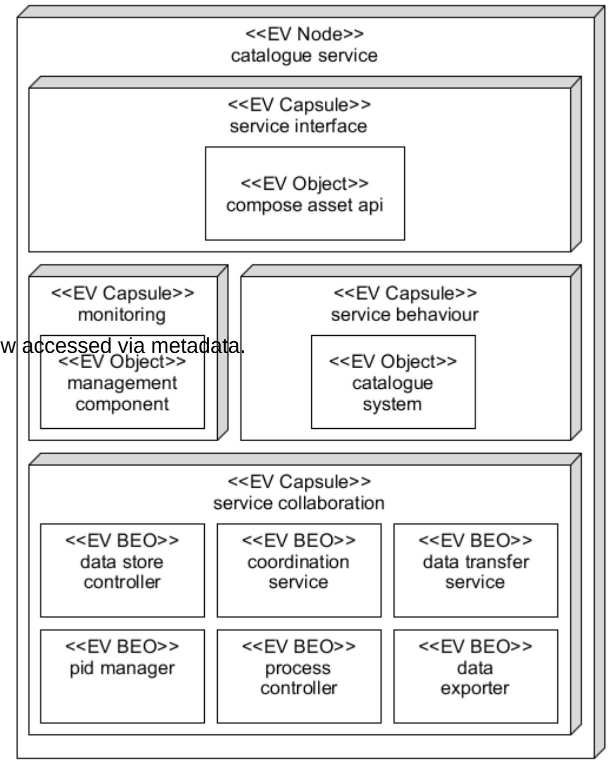
Name:	Move Asset
Process:	Move (selected parts of) asset to location accessed via metadata
Resources	• asset pid (precondition: discovery) • filtering parameters (metadata, optional)
Outputs	• asset (partial or completely relocated) • metadata • asset pid • asset reference (publishable) • asset description
Options	• move complete (delete original) • move partial (delete partial)

Compose Asset Configuration

The configuration for composing existing assets depends on seven BEOs: **catalogue service**, **data transfer service**, **data exporter**, **pid manager**, **data store controller**, **coordination service**, and **process controller**,

The figures on the right shows the configuration of the service using the recommended **Container Structure**.

This configuration shows a self contained service which already contains the required objects for supporting the compose asset process.



Name:	Compose Asset
Process:	Compose (selected parts of) asset into workflow accessed via metadata.
Resources	• asset pid (precondition: discovery) • filtering parameters (metadata, optional) • destination parameters • scheduling parameters
Outputs	• asset (copy, partial or complete) • metadata • asset pid • asset reference (publishable) • asset description
Options	• compose complete • compose partial

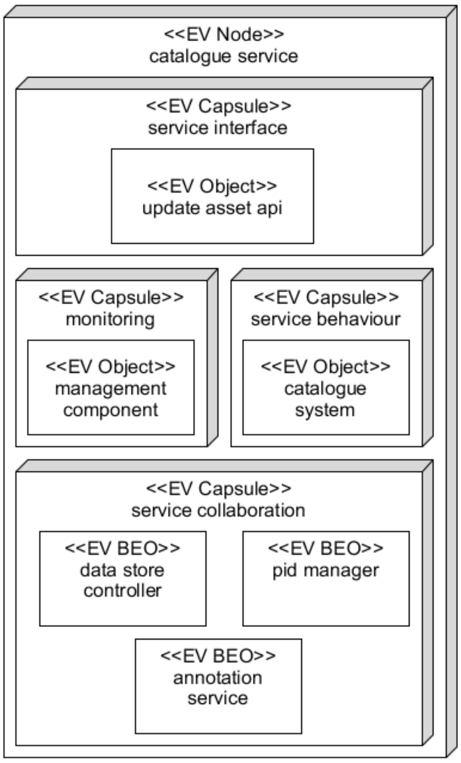
Update Asset Configuration

The configuration for discovering an asset depends on four BEOs: **catalogue service**, **annotation service**, **pid manager**, and **data store controller**.

The figures on the right shows the configuration of the service using the recommended **Container Structure**.

This configuration shows a self contained service which already contains the required objects for supporting the discovery process.

Name:	Discover Asset
Process:	Update an asset and/or metadata
Resources	• asset pid • update parameters (metadata) • asset (complement or replacement)
Outputs	• metadata • asset pid • asset reference (publishable) • asset description
Optionsup	• update metadata only • annotate asset • annotate metadata • update global pid



RM ODP Background (Draft)

This section provides a brief introduction of the main concepts that support the development of the engineering viewpoint [01]. The main ODP concepts for the definition of the engineering viewpoint are the **Object Types** and the **Architectural Model**.

[01] For further details refer to the ODP standard [4] and the book from Linington et. al. [37]

EV Object Types (Draft)

Following the ODP model, there are four types of Engineering Viewpoint objects: **Engineering Objects** (EV Objects) **Basic Engineering Objects** (EV BEO), **Container Objects** (EV Container) and **Channel Objects** (EV Channel). EV Objects provide functionalities which enable BEO distribution, EV BEOs map one to one with CV objects from the CV, EV Containers are used to group EV Objects and EV BEOs, and EV Channel objects are used to connect EV BEOs across containers [1].

Engineering Objects

An engineering object (EV object) is any object of interest in this viewpoint. For this reason BEOs, **Container Objects**and **Channel Objects**are also engineering objects (specialisations of). EV objects support computational requirements, distribution transparencies or infrastructure aspects of the system. The main distinction is made between BEOs, which represent computational objects (**Computational Viewpoint**), and other engineering objects, whose aim is to provide basic engineering functions, such as managers, interceptors and directories [2]

In the ENVRI RM, **Computational Viewpoint** are generic and can be used by any subsystem. In the EV, **Computational Viewpoint** can be specialisations to better describe a fully working subsystem. In practice, RIs may already have defined their systems and the division between subsystems may not be obvious (or as strictly defined as in the RM). In this case the subsystems will be viewed as logical not physical grouping of objects. This decomposition (logical or physical) can be used to identify practical interfaces for inter-operating between subsystems in different RIs. The different types of EV Containers allow the definition of both physically distributed and logically distributed containers. Thus, subsystems can be tightly coupled or extremely loose. The definition of EV channels will depend on the distribution model selected.

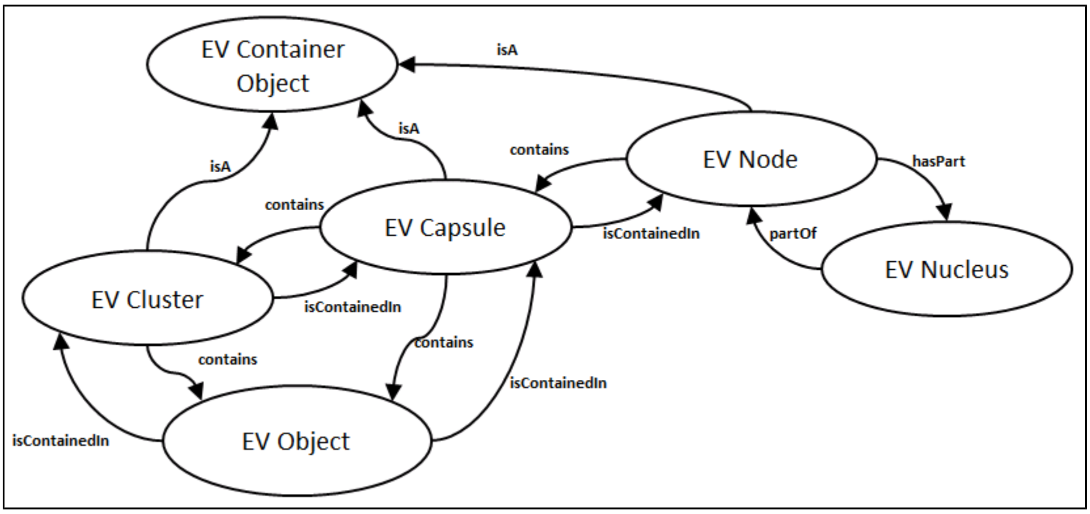
Basic Engineering Objects

A Basic Engineering Object (BEO) is a special kind of **Engineering Objects**, which is used to give a representation of a computational object in the engineering viewpoint. The EV is concerned primarily with the engineering of machines and of network communications. Some computational objects may represent human actors, but for these there is just a simple placeholder BEO; the engineering of communication with them is a matter for HCI standardisation, but is not detailed in the reference model and so it is not discussed further here [3].

The set of BEOs can be seen as abstractions of the computational design. The resulting description hides distinctions between objects with similar communications requirements, and retaining only the information about the computational objects that characterizes them as users of the distribution platform being provided. Therefore, the BEO is the primary object to be placed on a particular node, and which initiates communication across the network. All other engineering objects are secondary elements defined in an **Architectural Model**, whose goal is to provide the functions necessary to support distribution. This includes a variety of supporting objects, like repositories or directories, which are drawn from a set of common functions called ODP functions [Footnote03].

For instance, if a component is exposed, it becomes a BEO, but if it is kept private then it is not part of the EV.

Container Objects



The engineering viewpoint language defines four types of container objects to describe the principal controlling elements involved in any engineering specification: node, nucleus, capsule and cluster [4]. The figure to the right shows the containment relationships between containers, and engineering objects.

A **node** represents a physical object that has computing, communication and storage capabilities, and generally has connections to other nodes. A node will, therefore, have one or more network addresses, and the elements deployed on the node can become network accessible. A laptop, a PC, and server machines are all nodes. However, this is a slight simplification because a node does not actually have to be a separate, physically tangible object; a virtualization of such a resource can also be considered to be a node (a virtual machine for instance) [Footnote04].

A **nucleus** represents the basic mechanisms needed to make a node function at the lowest level, typically representing an operating system kernel that manages and allocates processing capabilities, communication capabilities and storage capabilities. Fair scheduling and accurate timing both depend on the centralised control of the node's resources offered by the nucleus [Footnote04].

A **capsule** is a unit of independent processing and storage. Faults within a capsule can affect all of the objects in it. The capsule supports a collection of engineering objects managed by a capsule manager. Capsules are isolated from one another by some protection mechanism, so that incorrect behaviour in one capsule does not damage other capsules. One of the consequences of this is typically that, because of the extra checking involved, communication across capsule boundaries is much more expensive than communication within a capsule. An example of a set of capsules is the set of independent processes, each with its own address space, run by an operating system (a nucleus). Another example of a capsule is a JEE or CCM component container [Footnote04].

A **cluster** is a collection of **Basic Engineering Objects** that have closely coupled lifecycles, and so can be activated, deactivated or migrated as one single unit. The record of objects that makes up a JEE or .NET component is a cluster. Another example is the aggregation of small primitive objects into a larger configuration to form a row for database update [Footnote04].

Channel Objects

The engineering viewpoint is concerned with defining a channel architecture that represents the communication infrastructure, which allows engineering objects to interact. The basic element is the channel, which is the engineering equivalent of a computational binding. A channel consists of stubs, binders, protocol objects and interceptors, communicating **Basic Engineering Objects**, generally residing in different nodes [5].

Stubs transform or monitor information in the channel. This includes, for example, the marshalling and unmarshalling of message elements, the translation of local interfaces into interoperable interface references, or provision of message content encryption. Stubs are the elements that enable access transparency in the communication between two objects written in different languages (such as C++ and smalltalk). The client object talks to its local stub, which is in charge of translating the request into a neutral format that is sent along the channel and that the server stubs understand. The received request is then translated by the server stub to the language of the server, and passed to it. The response from the server follows a similar route back to the client, with the stubs again translating the messages. The result is that the client and server objects both think that they are talking to local objects written in their own language [Footnote05].

Binders provide services to establish a distributed binding between the BEOs communicating through the channel and to provide the transparency functions that coordinate replicated object instances. There can be a number of different dialogue styles involved in this. Thus, for instance, the client and server binders can set up the communication channel and the server binder can wait for requests before activating the server object. In fact, the server binder can exhibit different behaviours depending on the activation policy required for the server object; the binder can create one object for every request received, or instantiate only one object to take care of all incoming requests when the channel is started, or it can create one server object that takes care of all requests received during a period of time, but which terminates if it receives no requests for a while; many other instantiation policies are possible [Footnote05].

A **protocol** object is an encapsulation of the communication capability of the protocols, which may be a full stack of layered protocols for a specific task, such as support for the Web Services protocol SOAP. At a lower level, protocol objects might exploit IPv6 roaming support to keep in contact with mobile devices in a way that does not involve the recreation of bindings. A protocol object may also encapsulate an implementation of some special purpose protocol, such as a driver for a noise resistant satellite link or a quantum cryptography channel [Footnote05].

Interceptors are the elements provided by the engineering viewpoint to provide a gatewae when there is a need to cross organizational, security, system management, naming, or protocol domain boundaries. The fact that all the messages exchanged go through the channel interceptor also enables the addition of interesting management functionality to the system. For instance, messages can be observed in order to carry out quality of service and performance monitoring, or even reordered or filtered for security or other reasons. Another use is where two objects live in different networks, each following some local communication protocol; one uses OSI's seven layers architecture and the other one follows a vendor's proprietary protocol, for instance. The task of bridging these differences can be carried out by an interceptor within the engineering channel that connects them. Interceptors need not analyse all the layers of encapsulation in the communication; they can just pass on information unchanged if it is already understandable to both sides [Footnote05].

The engineering viewpoint defines a referece architecture. This Architecture is defined in three parts: the component model, and the way the components are integrated. The integration of components is defined as the node architecture and the channel architecture. The

[1] See D5.2 p. 48. [42]

[2] See Linington et.al. p94 [37]

[3] See Linington et.al. p. 91-92 [37]

[4] See Linington et.al. p. 94-95 [37]

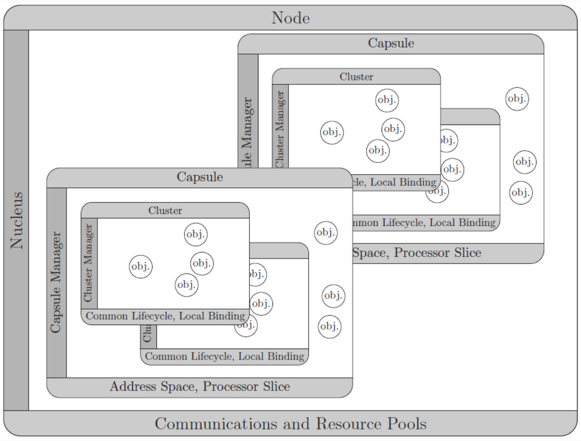
[5] See Linington et.al. p. 96-98 [37]

EV Architectural Model (Draft)

The main value of the engineering viewpoint is in providing a technology neutral architectural framework or reference architecture that can be used as a basic tool for designing new systems or comparing existing infrastructure technologies for distributed processing[1]. This description includes the basic objects that are composed to provide the system functionalities, the structural components that make this integration possible and the basic architecture of nodes (**Node Architecture**) and the architecture of the communication structures that connect those nodes (**Channel Architecture**).

Node Architecture

The node architecture identifies a number of functions that are commonly found in the platforms supporting application objects. Providing a technology neutral architecture for describing them helps to identify the commonalities between different infrastructure solutions, such as JEE or .NET, and so makes the integration of different platforms simpler[2]. Several of these functions are concerned with groupings of objects that are important when managing essential system properties, such as resource consumption, fault tolerance or persistence. These groupings, include such elements as nodes and, within them, sets of clusters, with cluster managers, sets of capsules, with capsule managers, and a nucleus.


Node
Capsule
Cluster
Cluster Manager
Capsule Manager
obj.
Common Lifecycle, Local Binding
Address Space, Processor Slice
Nucleus
Communications and Resource Pools

Generic Node Structure (From Linington et. al. [37])

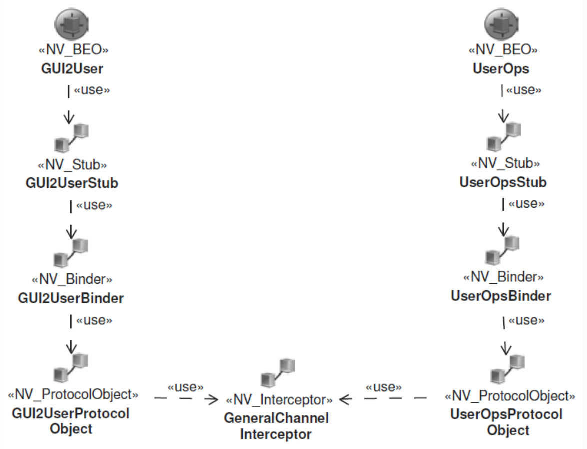
It is important to understand the kinds of functionality required to support provision of distribution transparencies, the basic mechanisms needed to achieve those functionalities, and how they can be structured and assigned to the architectural elements of the engineering viewpoint. This provides a reference architecture, against which existing distributed processing systems can be compared and evaluated, or with which tomorrow's systems can be designed. Using it helps technical architects specify or select required infrastructure characteristics[2].

Channel Architecture

The engineering viewpoint is also concerned with defining a channel architecture that represents the communication infrastructure, which allows engineering objects to interact (across container barriers)[3].

All channels involve the same functions, using specialized engineering objects to implement the required functionality in an ordered manner. However, different specific communication architectures may each organize or interleave these functions in their own way[3].

Normally there is no need to specify these elements in detail because they are provided by the underlying middleware platform (some of which are themselves standardized). However, there are occasions in which we want to model them functionally, in order to specify some of a channel's properties, or to express requirements on it - such as performance requirements on the channels or security constraints on the protocol objects or interceptors[3].



Example of a Channel Structure (From Linington et. al. [37])

[1] See Linington et.al. p91 [37]

[2] See Linington et.al. p. 93-96 [37]

[3] See Linington et.al. p. 96-98 [37]

Technology Viewpoint (Draft)

The Technology Viewpoint covers real-world constraints (such as restrictions on the facilities and technologies available to implement the system) applied to the existing computing platforms on which the computational processes must execute.

The specification of the Engineering viewpoint (EV) in the ENVRI RM aims to provide a flexible mapping between the models provided by the other viewpoints (**Science Viewpoint**, **Information Viewpoint**, **Computational Viewpoint**, **Engineering Viewpoint**) and the technologies and standards to be adopted for implementing them.

In the specification of the Technology Viewpoint for ENVRI, the engineering viewpoint configurations are mapped to an technology viewpoint objects. Each TV Object is represented as an API definition. The API definitions are provided as OpenAPI templates, which is the implementable standard for APIs, in this case the APIs correspond to generic implementations which can be further customised and OpenAPI is the implementable standard for APIs which can be used to generate template stubs for services and client libraries. Finally, the testing information is also included in the API specifications.

This compact definition of the Technology Viewpoint proposes the use of APIs as the main integration technology between RIs. By leaving the definition of the TV at this level we aim to avoid prescribing implementation technologies for core services and instead concentrate on technologies for RI distribution and integration.

The technology viewpoint defines: technology objects, implementable standards, implementation, conformance points, and IXIT (Implementation eXtra Information for Testing – a name derived from the earlier protocol specific OSI concept of a PIXIT).

Engineering Viewpoint components and their relationships

Catalogue Discovery Object (Draft)

The catalogue discovery object allows clients to discover resources registered in a catalogue, by providing three operations named queryCatalogue, catalogueDescription, and catalogueDomain. This object is always implemented by catalogue service implementations.

Name:	Catalogue Discovery
Description:	Discover the metadata available for a data asset or a set of assets
Correspondence:	Engineering Viewpoint: **discover asset configuration**

Operations:	**queryCatalogue** This operation enables formulating a query against a catalogue and return the url to the query result or table formatted output (depending on the size of the result) **describeCatalogue** Allows clients to retrieve a description or reference to the structure (schema), queryables, element sets, and formats of the metadata used for one or more registered resource types. **catalogueDomain** Allows clients to retrieve domain values for the different metadata elements
API Template:	**discovery_service_api.yaml**
Based on	Discovery class of the **OGC Catalogue Services 3.0 - General Model**
Example technologies	**pycsw**: an OGC CSW server implementation written in Python **CKAN**: CKAN can harvest CSW data when importing records and includes a CSW server to present a CSW interface to its own data.
Support in other models	**Australia's**: National Environmental Information Infrastructure: Reference Architecture

Conclusions and Future Work

The ENVRI Reference Model is a work in progress. Currently, attention is focused on three of the five ODP viewpoints: enterprise, information and computational. The remaining viewpoints of engineering and technology have been deferred to a later date.

Much work remains. Stronger correspondence between the three primary viewpoints is necessary to ensure that the three sub-models are synchronised in concept and execution. Further refactoring of individual components and further development of individual elements is to be expected as well. Further development of the presentation of the model is also essential, in order to both improve clarity to readers not expert in ODP and in order to promote a coherent position. In the immediate next step, the following tasks are planned:

Validation

The reference model will be validated from several aspects.

1. Usability. The users from different RIs will be invited to use the reference model to describe the research infrastructures in the ENVRI. The feedback will be collected and analysed to improve the definition of the reference model.
2. Interoperability. The descriptions of different RIs will be compared and check the commonality of the operations, and validate the effectiveness of the reference model in realizing the interoperability between RIs. The development of the use case in the work package 4 will also be used as the scenario to test the reference model.
3. Application. The linking model and the reference model will be tested in the application planning systems to check the data, resource and infrastructure interoperability

Semantic linking model

The reference model will be used as an important input for the development of semantic linking model among the reference model, data and infrastructure. The linking model provides an information framework to glue different information models of resources and data. The RM couples the semantic description of architectures and provides semantic interoperability between model descriptions. It needs to address fault tolerance, optimization and scheduling of linked resources, while making a trade-off between fuzzy logic and full information. The linking model is part of the development effort of the reference model.

The model is structured to support the semantic interoperability between data (data objects, metadata and annotations) which is provided by semantic mediation (or mapping, or translation) between descriptions of data (units, parameter, methods and others) and by semantic mediation of nominal and ordinal values, and/or taxonomies.

The linking model will take different aspects into considerations:

- The application (such as workflow) aspect captures the main characteristics of the application supported by the research infrastructure, including issues such as main flow patterns, quality of services, security and policies in user communities, and linking them to the descriptions of the data and infrastructures.

- The computing and data aspect focuses on operations and different data and meta data standards at different phase of data evolution (raw data, transfer, calibration, fusion etc.) and model them with linking of the data storing, accessing, delivery and etc. on (virtualized) e-Infrastructure.
- The Infrastructure aspect links the semantic model of the different layers of components in the physical infrastructure such as network elements and topologies, and also the monitoring information of the runtime status of the infrastructure. This part will enable the constraint solving of quality constraints to reserve and allocating resources for high level applications (processes).

Appendix A Common Requirements of Environmental Research Infrastructures

The following tables describe the common requirements environmental research infrastructures. The requirements are divided in five sets that correspond to the five stages of the datalifecycle. The requirements highlighted on each table are the minimal model.

Data Acquisition (A)

	Functions	**Definitions**
A.1	Instrument Integration	Functionality that creates, edits and deletes a sensor.
A.2	Instrument Configuration	Functionality that sets-up a sensor or a sensor network.
A.3	Instrument Calibration	Functionality that controls and records the process of aligning or testing a sensor against dependable standards or specified verification processes.
A.4	Instrument Access	Functionality that reads and/or updates the state of a sensor.
A.5	Configuration Logging	Functionality that collects configuration information or (run-time) messages from a sensor (or a sensor network) and outputs into log files or specified media which can be used by routine troubleshooting and in incident handling.
A.6	Instrument Monitoring	Functionality that checks the state of a sensor or a sensor network which can be done periodically or when triggered by events.
A.7	Parameter/Data Visualisation	Functionality that outputs the values of parameters and measured variables a display device.
A.8	*Real-Time Parameter/Data Visualisation*	*Specialisation of (Parameter) Visualisation which is subject to a real-time constraint.*
A.9	Process Control	Functionality that receives input status, applies a set of logic statements or control algorithms, and generates a set of analogue / digital outputs to change the logic states of devices.
A.10	Data Collection	Functionality that obtains digital values from a sensor instrument, associating consistent timestamps and necessary metadata.
A.11	*Real-Time Data Collection*	*Specialisation of Data Collection which is subject to a real-time constraint.*
A.12	Data Sampling	Functionality that selects a subset of individuals from within a statistical population to estimate characteristics of the whole population.
A.13	Noise Reduction	Functionality that removes noise from scientific data.
A.14	Data Transmission	Functionality that transfers data over communication channel using specified network protocols.

A.15	*Real-Time Data Transmission*	*Specialisation of Data Transmission which handles data streams using specified real-time transport protocols.*
A.16	Data Transmission Monitoring	Functionality that checks and reports the status of data transferring process against specified performance criteria.

Data Curation (B)

	Functions	**Definitions**
B.1	Data Quality Checking	Functionality that detects and corrects (or removes) corrupt, inconsistent or inaccurate records from data sets.
B.2	Data Quality Verification	Functionality that supports manual quality checking.
B.3	Data Identification	Functionality that assigns (global) permanent unique identifiers to data products.
B.4	Data Cataloguing	Functionality that associates a data object with one or more metadata objects which contain data descriptions.
B.5	Data Product Generation	Functionality that processes data against requirement specifications and standardised formats and descriptions. (optional/may be null)
B.6	Data Versioning	Functionality that assigns a new version to each state change of data, allows to add and update some metadata descriptions for each version, and allows to select, access or delete a version of data.
B.7	Workflow Enactment	Functionality that interprets predefined process descriptions and control the instantiation of processes and sequencing of activities, adding work items to the work lists and invoking application tools as necessary.
B.8	Data Storage & Preservation	Functionality that deposits (over long-term) the data and metadata or other supplementary data and methods according to specified policies, and makes them accessible on request.
B.9	Data Replication	Functionality that creates, deletes and maintains the consistency of copies of a data set on multiple storage devices.
B.10	Replica Synchronisation	Functionality that exports a packet of data from on replica, transports it to one or more other replicas and imports and applies the changes in the packet to an existing replica.

Data Publishing (C)

	Functions	**Definitions**
C.1	Access Control	Functionality that approves or disapproves of access requests based on specified access policies.
C.2	Resources Annotation	Functionality that creates, changes or deletes a note that reading any form of text, and associates them with a computational object.

C.3	*Data Annotation*	*Specialisation of Resource Annotation which allows to associate an annotation to a data object.*
C.4	Metadata Harvesting	Functionality that (regularly) collects metadata (in agreed formats) from different sources.
C.5	Resource Registration	Functionality that creates an entry in a resource registry and inserts resource object or a reference to a resource object in specified representations and semantics.
C.6	*Metadata Registration*	*Specialisation of Resource Registration, which registers a metadata object in a metadata registry.*
C.7	*Identifier Registration*	*Specialisation of Resource Registration, which registers an identifier object in an identifier registry.*
C.8	*Sensor Registration*	*Specialisation of Resource Registration which registers a sensor object to a sensor registry.*
C.9	Data Conversion	Functionality that converts data from one format to another format.
C.10	Data Compression	Functionality that encodes information using reduced bits by identifying and eliminating statistical redundancy.
C.11	Data Publication	Functionality that provides clean, well-annotated, anonymity-preserving datasets in a suitable format, and by following specified data-publication and sharing policies to make the datasets publicly accessible or to those who agree to certain conditions of use, and to individuals who meet certain professional criteria.
C.12	Data Citation	Functionality that assigns an accurate, consistent and standardised reference to a data object, which can be cited in scientific publications.
C.13	Semantic Harmonisation	Functionality that unifies similar data (knowledge) models based on the consensus of collaborative domain experts to achieve better data (knowledge) reuse and semantic interoperability.
C.14	Data Discovery and Access	Functionality that retrieves requested data from a data resource by using suitable search technology.
C.15	Data Visualisation	Functionality that displays visual representations of data.

Data Processing (D)

	Functions	**Definitions**
D.1	Data Assimilation	Functionality that combines observational data with outputs from a numerical model to produce an optimal estimate of the evolving state of the system.
D.2	Data Analysis	Functionality that inspects, cleans, and transforms data, providing data models which highlight useful information, suggest conclusions, and support decision making.

D.3	Data Mining	Functionality that supports the discovery of patterns in large data sets.
D.4	Data Extraction	Functionality that retrieves data out of (unstructured) data sources, including web pages ,emails, documents, PDFs, scanned text, mainframe reports, and spool files.
D.5	Scientific Modelling and Simulation	Functionality that supports the generation of abstract, conceptual, graphical or mathematical models, and to run an instances of those models.
D.6	*Scientific Workflow Enactment*	*Functionality provided as a specialisation of Workflow Enactment supporting the composition and execution of computational or data manipulation steps in a scientific application. Important processing results should be recorded for provenance purposes.*
D.7	Scientific Visualisation	Functionality that graphically illustrates scientific data to enable scientists to understand, illustrate and gain insight from their data. (optional or may be null)
D.8	Service Naming	Functionality that encapsulates the implemented name policy for service instances in a service network.
D.9	Data Processing Control	Functionality that initiates calculations and manages the outputs to be returned to the client.
D.10	Data Processing Monitoring	Functionality that checks the states of a running service instance.

Data Use (E)

	Functions	Definitions
E.1	Authentication	Functionality that verifies a credential of a user.
E.2	Authorisation	Functionality that specifies access rights to resources.
E.3	Accounting	Functionality that measures the resources a user consumes during access for the purpose of capacity and trend analysis, and cost allocation.
E.4	*User Registration*	*Specialisation of Resource Registration which registers a user to a user registry.*
E.5	Instant Messaging	Functionality for quick transmission of text-based messages from sender to receiver.
E.6	Interactive Visualisation	Functionality that enables users to control of some aspects of the visual representations of information.
E.7	Event Notification	Functionality that delivers message triggered by predefined events.

Appendix B Terminology and Glossary

Acronyms and Abbreviations

CCSDS	Consultative Committee for Space Data Systems
CMIS	Content Management Interoperability Services
CERIF	Common European Research Information Format
DDS	Data Distribution Service for Real-Time Systems
ENVRI	Environmental Research Infrastructure
ENVRI_RM	ENVRI Reference Model
ESFRI	European Strategy Forum on Research Infrastructures
ESFRI-ENV RI	ESFRI Environmental Research Infrastructure
GIS	Geographic Information System
IEC	International Electrotechnical Commission
ISO	International Organisation for Standardization
OAIS	Open Archival Information System
OASIS	Organization for the Advancement of Structured Information Standards
ODP	Open Distributed Processing
OGC	Open Geospatial Consortium
OMG	Object Management Group
ORCHESTRA	Open Architecture and Spatial Data Infrastructure for Risk Management
ORM	OGC Reference Model
OSI	Open Systems Interconnection
OWL	Web Ontology language
SOA	Service Oriented Architecture
SOA-RM	Reference Model for Service Oriented Architecture
RDF	Resource Description Framework
RM-OA	Reference Model for the ORCHESTRA Architecture
RM-ODP	Reference Model of Open Distributed Processing
UML	Unified Modelling Language
W3C	World Wide Web Consortium
UML4ODP	Unified Modelling Language For Open Distributed Processing

Terminology

Access Control: A functionality that approves or disapproves of access requests based on specified access policies.

Acquisition Service: Oversight service for integrated data acquisition.

Active role: A active role is typically associated with a human actor.

Add Metadata: Add additional information according to a predefined schema (metadata schema). This partially overlaps with data annotations.

Annotate Data: Annotate data with meaning (concepts of predefined local or global conceptual models).

Annotate Metadata: Link metadata with meaning (concepts of predefined local or global conceptual models). This can be done by adding a pointer to concepts within a conceptual model to the data. If e.g. concepts are terms in and SKOS/RDF thesaurus, published as linked data then this would mean entering the URL of the term describing the meaning of the data.

Annotation: (verb) The action of annotating or making notes. (noun) A note added to anything written, by way of explanation or comment.

Annotation Service: Oversight service for adding and updating records attached to curated datasets.

Assign Unique Identifier: Obtain a unique identifier and associate it to the data.

Authentication: A functionality that verifies a credential of a user.

Authentication Service: Security service responsible for the authentication of external agents making requests of infrastructure services.

Authorisation: A functionality that specifies access rights to resources.

Authorisation Service: Security service responsible for the authorisation of all requests made of infrastructure services by external agents.

Backup: A copy of (persistent) data so it may be used to restore the original after a data loss event.

Behaviour : A behaviour of a community is a composition of actions performed by roles normally addressing separate business requirements.

Build Conceptual Models: Establish a local or global model of interrelated concepts.

Capacity Manager: An active role, which is a person who manage and ensure that the IT capacity meets current and future business requirements in a cost-effective manner.

Carry out Backup: Replicate data to an additional data storage so it may be used to restore the original after a data loss event. A special type of backup is a long term preservation.

Catalogue service: Oversight service for cataloguing curated datasets.

Check Quality: Actions to verify the quality of data.

Citation: from the ENVRI RM perspective, citation is defined as a pointer from a publication to:

- data source(s)
- and/or the owner(s) of the data source(s)
- a description of the evaluation process, if available
- a timestamp marking the access time to the data sources, thus reflecting a certain version

Citizen (synonyms: General Public, Media): An active role, a person, who is interested in understanding the knowledge delivered by an environmental science research infrastructure, or discovering and exploring the Knowledge_Base_Glossary enabled by the research infrastructure.

Citizen Scientist: An active role, member of the general public who engages in scientific work, often in collaboration with or under the direction of professional scientists and scientific institutions (also known as amateur scientist).

Community: A collaboration which consists of a set of roles agreeing their objective to achieve a stated business purpose.

Concept: Name and definition of the meaning of a thing (abstract or real thing). Human readable definition by sentences, machine readable definition by relations to other concepts (machine readable sentences). It can also be meant for the smallest entity of a conceptual model. It can be part of a flat list of concepts, a hierarchical list of concepts, a hierarchical thesaurus or an ontology.

Conceptual Model: A collection of concepts, their attributes and their relations. It can be unstructured or structured (e.g. glossary, thesaurus, ontology). Usually the description of a concept and/or a relation defines the concept in a human readable form. Concepts within ontologies and their relations can be seen as machine readable sentences. Those sentences can be used to establish a self-description. It is, however, practice today, to have both, the human readable description and the machine readable description. In this sense a conceptual model can also be seen as a collection of human and machine readable sentences. Conceptual models can reside within the persistence layer of a data provider or a community or outside. Conceptual models can be fused with the data (e.g. within a network of triple stores) or kept separately.

Coordination Service: An oversight service for data processing tasks deployed on infrastructure execution resources.

Data Acquisition Community. A community, which collects raw data and bring (streams of) measures into a system.

Data Acquisition Subsystem: A subsystem that collects raw data and brings the measures or data streams into a computational system.

Data Analysis: A functionality that inspects, cleans, transforms data, and provides data models with the goal of highlighting useful information, suggesting conclusions, and supporting decision making.

Data Assimilation: A functionality that combines observational data with output from a numerical model to produce an optimal estimate of the evolving state of the system.

Data Broker: Broker for facilitating data access/upload requests**.**

Data Cataloguing: A functionality that associates a data object with one or more metadata objects which contain data descriptions.

Data Citation: A functionality that assigns an accurate, consistent and standardised reference to a data object, which can be cited in scientific publications.

Data Collection: A behaviour performed by a *Data Collector* that control and monitor the collection of the digital values from a *sensor* instru ment or a human sensor, such as a *Measurer* or a *Observer*, associating consistent time-stamps and necessary metadata.

Data Collector: Active or passive role, adopted by a person or an instrument collecting data.

Data Consumer: Either an active or passive role, which is an entity who receives and use the data.

Data Curation Community: A community, which curates the scientific data, maintains and archives them, and produces various data products with metadata.

Data Curation Subsystem: A subsystem that facilitates quality control and preservation of scientific data.

Data Curator: An active role, which is a person who verifies the quality of the data, preserve and maintain the data as a resource, and prepares various required data products.

Data Discovery & Access: A functionality that retrieves requested data from a data resource by using suitable search technology.

Data Exporter: Binding object for exporting curated datasets.

Data Extraction: A functionality that retrieves data out of (unstructured) data sources, including web pages, emails, documents, PDFs, scanned text, mainframe reports, and spool files.

Data Identification: A functionality that assigns (global) unique identifiers to data contents.

Data Importer: An Oversight service for the import of new data into the data curation subsystem.

Data infrastructure: a collection of data assets, organisations that operate and maintain them and guides describing how to use and manage the data. A data infrastructure is sustainably funded and has oversight that provides direction to maximise data use and value by meeting the needs of society. Data infrastructure includes technology, processes and organisation.

Data management: a process development and execution of architectures, policies, practices and procedures in order to manage the data lifecycle needs of a specific research community.

Data management plan (DMP): a formal document that outlines how data are to be handled both during a research project and after the project is completed.

Data Mining: A functionality that supports the discovery of patterns in large data sets.

Data Originator: Either an active or a passive role, which provide the digital material to be made available for public access.

Data Processing Control: A functionality that initiates the calculation and manages the outputs to be returned to the client.

Data Processing Subsystem: A subsystem that aggregates the data from various resources and provides computational capabilities and capacities for conducting data analysis and scientific experiments.

Data Product Generation: A functionality that processes data against requirement specifications and standardised formats and descriptions.

Data Provenance: Information that traces the origins of data and records all state changes of data during their lifecycle and their movements between storages.

Data Provider: Either an active or a passive role, which is an entity providing the data to be used.

Data Publication: A functionality that provides clean, well-annotated, anonymity-preserving datasets in a suitable format, and by following specified data-publication and sharing policies to make the datasets publically accessible or to those who agree to certain conditions of use, and to individuals who meet certain professional criteria.

Data Publication Community: A community that assists the data publication, discovery and access.

(Data Publication) Repository: A passive role, which is a facility for the deposition of published data.

Data Publishing Subsystem: A subsystem that enables discovery and retrieval of data housed in data resources.

Data Quality Checking: A functionality that detects and corrects (or removes) corrupt, inconsistent or inaccurate records from data sets.

Data Service Provision Community: A community that provides various services, applications and software/tools to link, and recombine data and information in order to derive knowledge.

Data State: Term used as defined in ISO/IEC 10746-2. At a given instant in time, data state is the condition of an object that determines the set of all sequences of actions (or traces) in which the object can participate.

Data Storage & Preservation: A functionality that deposits (over long-term) the data and metadata or other supplementary data and methods according to specified policies, and makes them accessible on request.

Data Store Controller: A data store within the data curation subsystem.

Data Transfer Service: Oversight service for the transfer of data into and out of the data curation subsystem.

Data Transmission: A functionality that transfers data over communication channel using specified network protocols.

Data Transporter: Generic binding object for data transfer interactions.

Data Use Community: A community who makes use of the data and service products, and transfers the knowledge into understanding.

Data Use Subsystem: A subsystem that provides functionalities to manage, control, and track users' activities and supports users to conduct their roles in the community.

Describe Service: Describe the accessibility of a service or process, which is available for reuse, the interfaces, the description of behaviour and/or implemented algorithms.

Design of Measurement Model: A behaviour that designs the measurement or monitoring model based on scientific requirements.

Do Data Mining: Execute a sequence of metadata / data request --> interpret result --> do a new request

e-Infrastructure: a combination and interworking of digitally-based technology (hardware and software), resources (data, services, digital libraries), communications (protocols, access rights and networks), and the people and organisational structures needed to support modern, internationally leading collaborative research be it in the arts and humanities or the sciences.

Educator (synonym: Trainer): An active role, which is a person who makes use of the data and application services for education and training purposes.

Engineer (synonym: Technologist): An active role, which is a person who develops and maintains the research infrastructure.

Environmental Scientist: An active role, which is a person who conduct research or perform investigation for the purpose of identifying, abating, or eliminating sources of pollutants or hazards that affect either the environment or the health of the population. Using knowledge of various scientific disciplines, may collect, synthesize, study, report, and recommend action based on data derived from measurements or observations of air, food, soil, water, and other sources.

ENVRI Reference Model: A common ontological framework and standards for the description and characterisation of computational and storage systems of ESFRI environmental research infrastructures.

Experiment Laboratory: Community proxy for conducting experiments within a research infrastructure.

Field Laboratory: Community proxy for interacting with data acquisition instruments.

Final review: Review the data to be published, which will not likely be changed again.

Free text annotation: to add a short explanation or opinion to a text or drawing (equivalent to the dictionary definition of annotation).

Instrument Controller: An integrated raw data source.

Knowledge Base: (1) A store of information or data that is available to draw on. (2) The underlying set of facts, assumptions, and rules which a computer system has available to solve a problem.

Knowledge infrastructure: robust networks of people, artifacts, and institutions that generate, share, and maintain specific knowledge about the human and natural worlds.

Mapping Rule: Configuration directives used for model-to-model transformation.

(Measurement Model) Designer: An active role, which is a person who design the measurements and monitoring models based on the requirements of environmental scientists.

Measurement result: Quantitative determinations of magnitude, dimension and uncertainty to the outputs of observation instruments, sensors (including human observers) and sensor networks.

Measurer: An active role, which is a person who determines the ratio of a physical quantity, such as a length, time, temperature etc., to a unit of measurement, such as the meter, second or degree Celsius.

Metadata: Data about data, in scientific applications is used to describe, explain, locate, or make it easier to retrieve, use, or manage an information resource.

Metadata Catalogue: A collection of metadata, usually established to make the metadata available to a community. A metadata catalogue has an access service.

Metadata Harvesting (Publishing Community Role): A behaviour performed by a metadata harvester to gather metadata from data objects in order to construct catalogues of the available information. A functionality that (regularly) collects metadata (in agreed formats) from different sources.

Metadata Harvester (Publishing Community Role): A passive role performed by a system or service collecting metadata to support the construction/selection of a global conceptual model and the production of mapping rules.

Metadata State:

- raw: are established metadata, which are not yet registered. In general, they are not shareable in this status
- registered: are metadata which are inserted into a metadata catalogue.
- published: are metadata made available to the public, the outside world. Within some metadata catalogues registered.

Passive Role: A passive role is typically associated with a non-human actor.

Perform Mapping: Execute transformation rules for values (mapping from one unit to another unit) or translation rules for concepts (translating the meaning from one conceptual model to another conceptual model, e.g. translating code lists).

Persistent Data: Term (data) used as defined in ISO/IEC 10746-2. Data is the representations of information dealt by information systems and users thereof. Data which are persistent (stored).

Perform Measurement or Observation: Measure parameter(s) or observe an event. The performance of a measurement or observation produces measurement results.

PID Generator: A passive role, a system which assigns persist global unique identifiers to a (set of) digital object.

PID Registry: A passive role, which is an information system for registering PIDs.

PID Service: External service for persistent identifier assignment and resolution.

Policy Maker (synonym: Decision Maker): An active role, a person, who makes decisions based on the data evidences.

Process Control: A functionality that receives input status, applies a set of logic statements or control algorithms, and generates a set of analogue / digital outputs to change the logic states of devices.

Process Controller: Part of the execution platform provided by the data processing subsystem.

Process Data: Process data for the purposes of:

- converting and generating data products
- calculations: e.g., statistical processes, simulation models
- visualisation: e.g., alpha-numerically, graphically, geographically

Data processes should be recorded as provenance.

Provenance: The pathway of data generation from raw data to the actual state of data.

Publish Data: Make data public accessible.

Publish Metadata: Make the registered metadata available to the public.

QA Notation: Notation of the result of a Quality Assessment. This notation can be a nominal value out of a classification system up to a comprehensive (machine readable) description of the whole QA process.

Quality Assessment (QA): Assessment of details of the data generation, including the check of the plausibility of the data. Usually the quality assessment is done by predefined checks on data and their generation process.

Query Data: Send a request to a data store to retrieve required data.

Query Metadata: Send a request to metadata resources to retrieve metadata of interests.

Observer: An active role, which is a person who receives knowledge of the outside world through the senses, or records data using scientific instruments.

Raw Data Collector: Binding object for raw data collection.

Reference Mode: A reference mode is an abstract framework for understanding significant relationships among the entities of some environment.

Register Metadata: Enter the metadata into a metadata catalogue.

Research Infrastructure: means facilities, resources and related services that are used by the scientific community to conduct top-level research in their respective fields and covers major scientific equipment or sets of instruments; knowledge-based resources such as collections, archives or structures for scientific information; enabling Information and Communications Technology-based infrastructures such as Grid, computing, software and communication, or any other entity of a unique nature essential to achieve excellence in research. Such infrastructures may be “single-sited” or “distributed” (an organised network of resources) [41].

Resource Registration: A functionality that creates an entry in a resource registry and inserts resource object or a reference to a resource object in specified representations and semantics.

Role : A role in a community is a prescribing behaviour that can be performed any number of times concurrently or successively.

Science Gateway: Community portal for interacting with an infrastructure.

Scientific Modelling and Simulation: A functionality that supports the generation of abstract, conceptual, graphical or mathematical models, and to run an instance of the model.

Scientist (synonym: Researcher): An active role, which is a person who makes use of the data and application services to conduct scientific research.

(Scientific) Workflow Enactment: A specialisation of Workflow Enactment, which support of composition and execution a series of computational or data manipulation steps, or a workflow, in a scientific application. Important processes should be recorded for provenance purposes.

Security Service: Oversight service for authentication and authorisation of user requests to the infrastructure.

Semantic Annotation: link from an information object (single datum, data set, data container) to a concept within a conceptual model, enabling the discovery of the meaning of the information object by human and machines.

Semantic Broker: Broker for establishing semantic links between concepts and bridging queries between semantic domains.

SV Community Behaviour: A behaviour enabled by a *Semantic Mediator* that unifies similar data (knowledge) models based on the consensus of collaborative domain experts to achieve better data (knowledge) reuse and semantic interoperability.

Semantic Laboratory: Community proxy for interacting with semantic models.

Semantic Mediator: A passive role, which is a system or middleware facilitating semantic mapping discovery and integration of heterogeneous data.

Sensor: A passive role, which is a converter that measures a physical quantity and converts it into a signal which can be read by an observer or by an (electronic) instrument.

Sensor Network: A passive role, which is a network consists of distributed autonomous sensors to monitor physical or environmental conditions.

Service: Service or process, available for reuse.

Service Consumer: Either an active or a passive role, which is an entity using the services provided.

Service Description: Services and processes, which are available for reuse, be it within an enterprise architecture, within a research infrastructure or within an open network like the Internet, shall be described to help avoid wrong usage. Usually such descriptions include the accessibility of the service, the description of the interfaces, the description of behavior and/or implemented algorithms. Such descriptions are usually done along service description standards (e.g. WSDL, web service description language). Within some service description languages, semantic descriptions of the services and/or interfaces are possible (e.g. SAWSDL, Semantic Annotations for WSDL)

Service Provider: Either an active or a passive role, which is an entity providing the services to be used.

Service Registry: A passive role, which is an information system for registering services.

Setup Mapping Rules: Specify the mapping rules of data and/or concepts.

Specification of Investigation Design: This is the background information needed to understand the overall goal of the measurement or observation. It could be the sampling design of observation stations, the network design, the description of the setup parameters (interval of measurements) and so on... It usually contains important information for the allowed evaluations of data. (E.g. the question whether a sampling design was done randomly or by strategy determines which statistical methods that can be applied or not).

Specification of Measurements or Observations: The description of the scientific measurement model which specifies:

- what is measured;
- how it is measured;
- by whom it is measured; and
- what the temporal design is (single /multiple measurements / interval of measurement etc.)

Specify Investigation Design: specify design of investigation, including sampling design:

- geographical position of measurement or observation (site) -- the selections of observations and measurement sites, e.g., can be statistical or stratified by domain knowledge;
- characteristics of site;
- · preconditions of measurements.

Specify Measurement or Observation: Specify the details of the method of observations/measurements.

Stakeholder (synonyms: Private Investor, Private Consultant): An active role, a person, who makes use of the data and application service for predicting market so as to make business decision on producing related commercial products.

Storage: A passive role, which is memory, components, devices and media that retain digital computer data used for computing for some interval of time.

Storage Administrator: An active role, which is a person who has the responsibilities to the design of data storage, tune queries, perform backup and recovery operations, raid mirrored arrays, making sure drive space is available for the network.

Store Data: Archive or preserve data in persistent manner to ensure continuing accessible and usable.

Subsystem: a set of capabilities that collectively are defined by a set of interfaces with corresponding operations that can be invoked by other subsystems. Subsystems can be executed independently, and developed and managed incrementally.

Technician: An active role, which is a person who develop and deploy the sensor instruments, establishing and testing the sensor network, operating, maintaining, monitoring and repairing the observatory hardware.

Track Provenance: Add information about the actions and the data state changes as data provenances.

Unique Identifier (UID): With reference to a given (possibly implicit) set of objects, a unique identifier (UID) is any identifier which is guaranteed to be unique among all identifiers used for those objects and for a specific purpose.

User Behaviour Tracking: A behaviour enabled by a Data Use Subsystem that to track the Users. User Behaviour Tracking is the analysis of visitor behaviour on a website. The analysis of an individual visitor's behaviour may be used to provide options or content that relates to their implied preferences; either during a visit or in the future visits. Additionally, it can be user to track content use and performance.

User Group Work Supporting: A behaviour enabled by a Data Use Subsystem that to support controlled sharing, collaborative work and publication of results, with persistent and externally citable PIDs.

User Profile Management: A behaviour enabled by a Data Use Subsystem that to support persistent and mobile profiles, where profiles will include preferred interaction settings, preferred computational resource settings, and so on.

User Working Space Management: A behaviour enabled by a Data Use Subsystem that to support work spaces that allow data, document and code continuity between connection sessions and accessible from multiple sites or mobile smart devices.

User Working Relationships Management: A behaviour enabled by a Data Use Subsystem that to support a record of working relationships, (virtual) group memberships and friends.

Virtual Laboratory: Community proxy for interacting with infrastructure subsystems.

Virtual Research Environment (VRE, sysnomyms:Science Gateway, Collaboratory, Digital Library, Inhabited Information Space, e-Infrastructure, Cyberinfrastructure): a web-based working environment tailored to serve the needs of a research community. A VRE is expected to provide an array of commodities needed to accomplish the research community’s goal(s); it is open and flexible with respect to the overall service offering and lifetime; and it promotes fine-grained controlled sharing of both intermediate and final research results by guaranteeing ownership, provenance and attribution.

Appendix C Notation

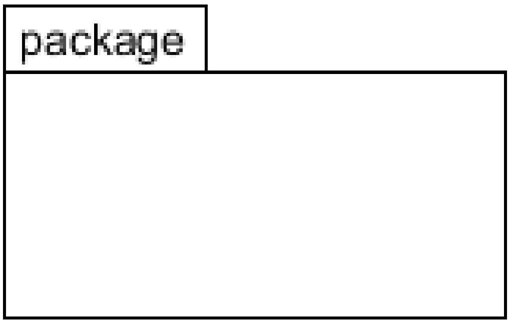
The notation used for the diagrams of the ENVRI RM is based on the UML notation suggested for ODP, the **UML4ODP** notation. The notation sections include a set of tables that describe the UML elements used to produce the diagrams presenting the different viewpoint models.

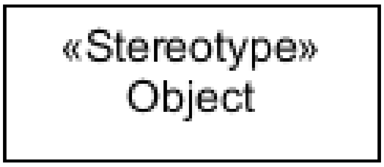
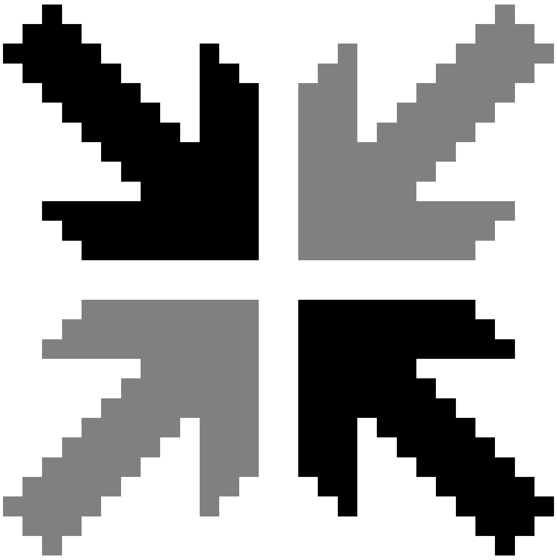
Notation of Science Viewpoint Models

Communities

SV communities are modelled using an object diagram. The following table describes the elements used in that diagram.

Table 1 Notation for community diagrams

Figure	Description
	A Package, in UML notation, is a grouping element. Package is used "to group elements, and to provide a namespace for the grouped elements". A package may contain other packages, thus providing for a hierarchical organization of packages. Classes, objects, use cases, components, nodes, node instances etc. can all be organized as packages, enabling a manageable organization of the elements of UML models.

	Objects are used to represent communities and objectives in the RM. The name refers to the represented entity(community or community objective). The stereotype indicates the namespace where the object is grouped. Sometimes the stereotype can be an image. The image can be used in place of the figure. The ODP stereotype for community is an icon of group of people with keyboards: The ODP stereotype for notation is an icon of group four black and grey arrows converging to a point:  Each community is linked to its corresponding objective by a line

In the following example diagram (Figure 1) the package represents an Environmental research infrastructure. The infrastructure contains five objects which are all communities and each community is linked to a community objective.

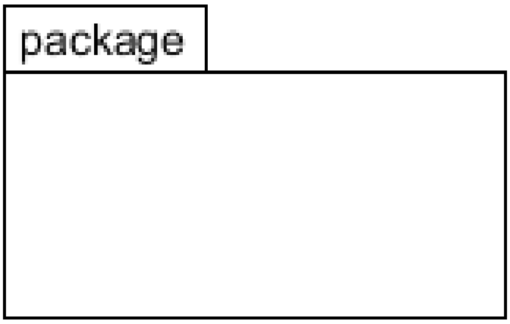
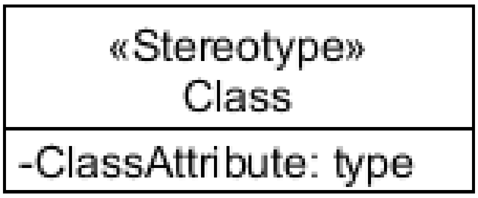
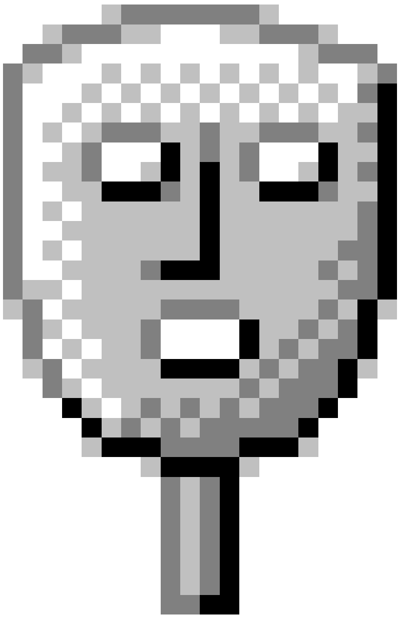
Figure 1 Example of a community diagram

Community Roles

SV Roles are represented using a class diagram with packages and classes

Table 2 Notation for role diagrams

Figure	Description

	A Package, in UML notation, is a grouping element. Package is used "to group elements, and to provide a namespace for the grouped elements". A package may contain other packages, thus providing for a hierarchical organization of packages. Classes, objects, use cases, components, nodes, node instances etc. can all be organized as packages, enabling a manageable organization of the elements of UML models.
	Classes are used to represent roles in the RM. Classes can have additional compartments to express properties (called attributes) and behaviours (called methods). Omitting the compartments means that the behaviour and attributes are undefined at the time of building the diagram. Name tag indicates the name of the class. Typically, classes are named using no spaces and starting each word that makes up the name, i.e., camelcase. The stereotype indicates the namespace where the class is grouped. Sometimes the stereotype can be an image. The image can be used in place of the figure. For ODP, the stereotype for role is a mask: 

In the example diagram the package represents the data curation community. The community contains eight role classes. The ENVRI RM provides a detailed description of each role in text.

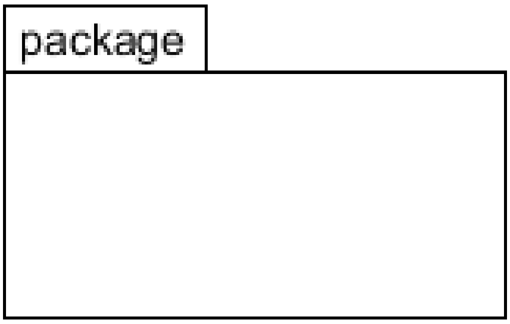
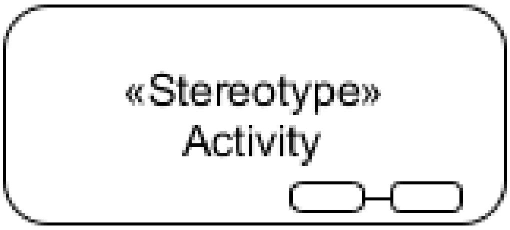
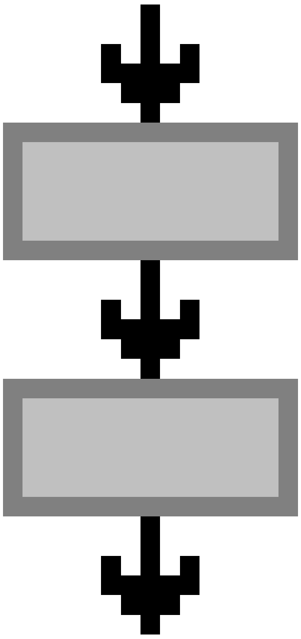
Figure 2 Example of a SV Role diagram

Community Behaviours

SV Behaviours are represented using an activity diagram with packages and activities

Table 3 Notation for behaviour diagrams

Figure	Description

 	A Package, in UML notation, is a grouping element. Package is used "to group elements, and to provide a namespace for the grouped elements". A package may contain other packages, thus providing for a hierarchical organization of packages. Classes, objects, use cases, components, nodes, node instances etc. can all be organized as packages, enabling a manageable organization of the elements of UML models.
 	Activities are used to represent behaviours in the RM. Name tag indicates the name of the behaviour. Behaviours are named using a short phrase that describes the event or action being represented. The small decoration in the activity indicates that the activity is complex and can be subdivided into smaller tasks. A stereotype can be used to indicate the namespace where the activity is grouped. Sometimes the stereotype can be an image. The stereotype image can be used in place of the figure. For ODP, the stereotype for behaviour is process icon: 

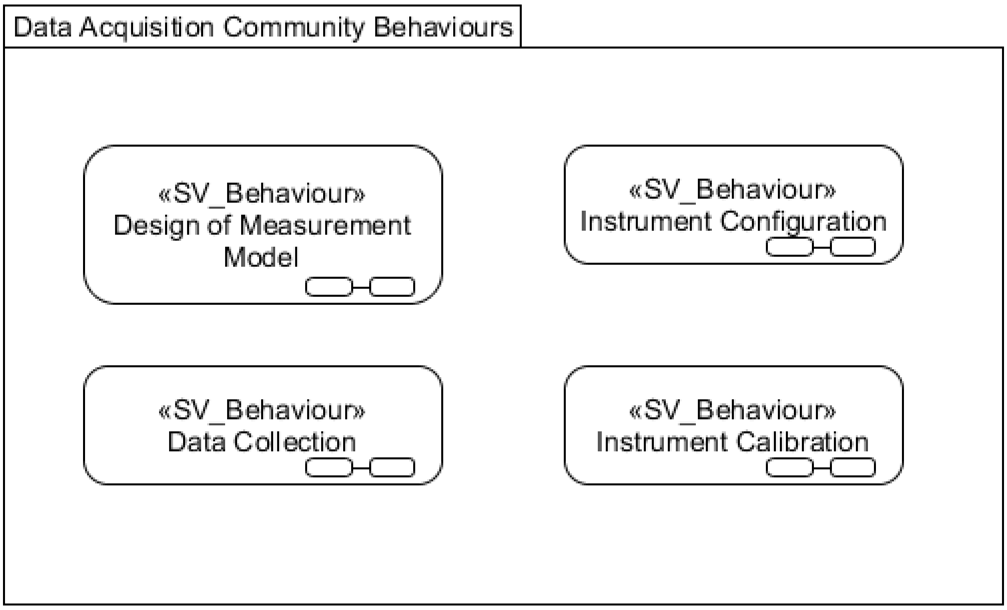

Data Acquisition Community Behaviours
«SV_Behaviour» Design of Measurement Model
«SV_Behaviour» Instrument Configuration
«SV_Behaviour» Data Collection
«SV_Behaviour» Instrument Calibration

Figure 3 Example of a SV Behaviour diagram

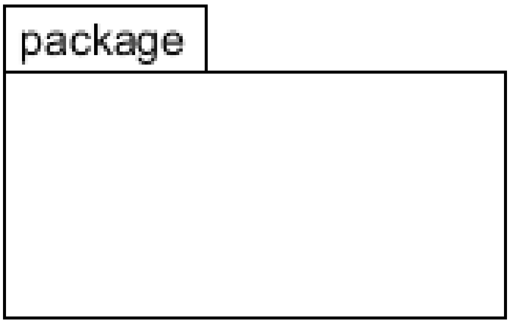
In the example diagram the package represents a community, data acquisition. The community implements four basic behaviours. The RM also provides a detailed description of each behaviour in text.

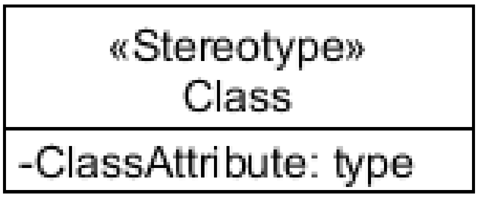
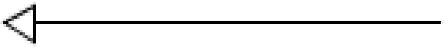
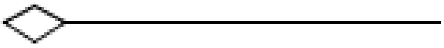
Notation of Information Viewpoint Models

Information Objects

IV Objects are represented using a class diagram.

Table 4 Notation for information object diagrams

Figure	Description
	A package, in UML notation, is a grouping element. Package is used "to group elements, and to provide a namespace for the grouped elements". A package may contain other packages, thus providing for a hierarchical organization of packages. Classes, objects, use cases, components, nodes, node instances etc. can all be organized as packages, enabling a manageable organization of the elements of UML models.

	Classes are used to represent information objects in the RM. Classes can have additional compartments to express properties (called attributes) and behaviours (called methods). Leaving the compartments blank means that the behaviour and attributes are undefined at the time of creating the diagram. Name tag indicates the name of the class. Typically, classes are named using no spaces in camelcase. The stereotype indicates the namespace where the class is grouped. Sometimes the stereotype can be an image. The image can be used in place of the figure. For ODP, the stereotype for information object is an “i” icon with a tag on top:
	Generalisation relationship indicates that one of the two related classes (the subclass) is considered to be a specialized form of the other (the super class). Generalisation is represented with an arc with a blank triangle decoration. The blank triangle points to the super class and the undecorated end is connected to the subclass. The generalization relationship is also known as the inheritance or "is a" relationship.
	Aggregation relationship indicates an association that represents a part-whole or part-of relationship. Aggregation is represented with an arc with a blank rhombus decoration. The blank rhombus shape indicates the composite and the undecorated end of the arc is the component.

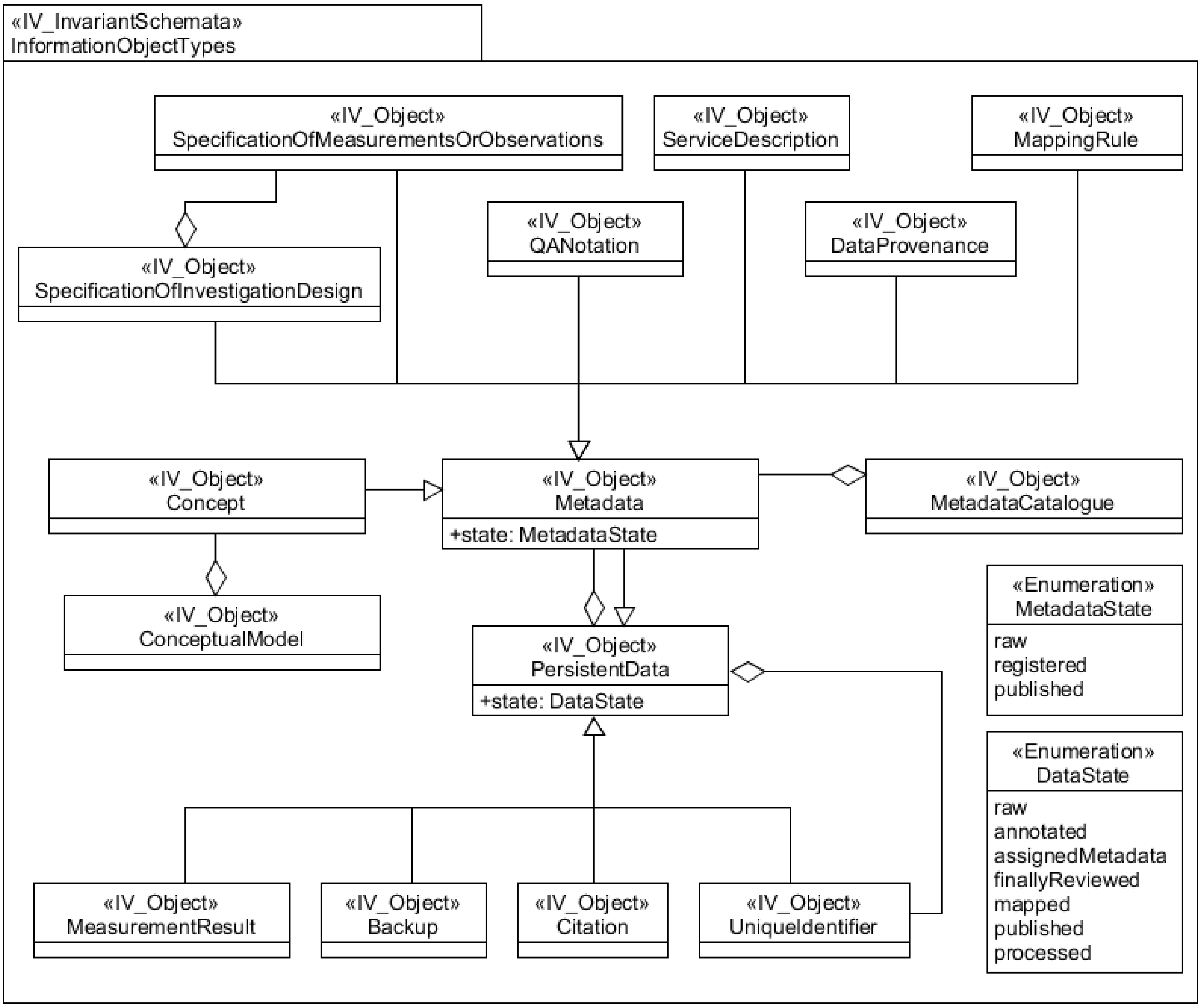


Figure 4 Example of an IV Object diagram

In the example diagram the package represents the collection of all information objects described by the ENVRI RM. The stereotype for the package is invariant schemata, which indicates that these are the parts of the model that are stable. The main objects are persistent data and metadata. The RM also provides a detailed description of each object in the text.

Information Actions

IV Actions are represented using an activity diagram with packages and activities

Table 5 Notation for action type diagrams

Figure	Description

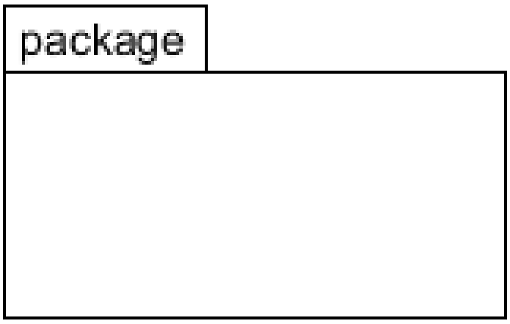
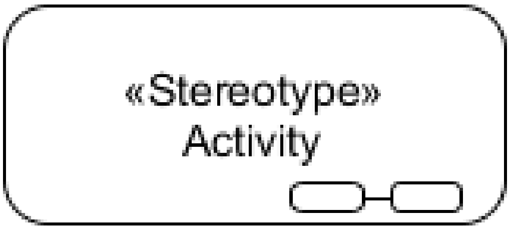
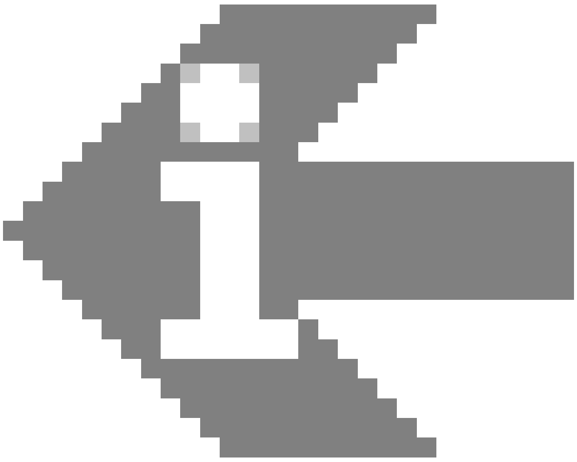
	A package, in UML notation, is a grouping element. Package is used "to group elements, and to provide a namespace for the grouped elements". A package may contain other packages, thus providing for a hierarchical organization of packages. Classes, objects, use cases, components, nodes, node instances etc. can all be organized as packages, enabling a manageable organization of the elements of UML models.
	Activities are used to represent action in the RM. Name tag indicates the name of the action. Actions are named using a short phrase that describes the event or action being represented. The small decoration in the box indicates that the action is complex and can be subdivided into smaller tasks. A stereotype can be used to indicate the namespace where the activity is grouped. Sometimes the stereotype can be an image. The stereotype image can be used in place of the figure. For ODP, the stereotype for information action is an arrow icon with a lowercase “i”: 

Figure 5 Example of an IV Action Types diagram

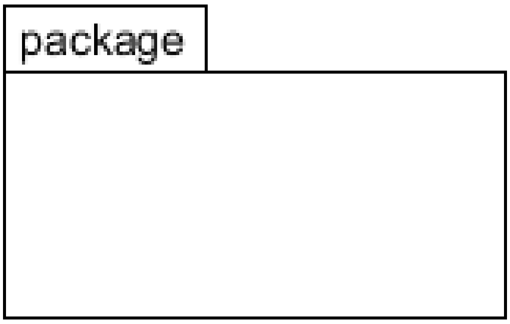
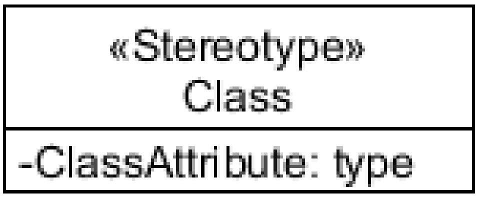
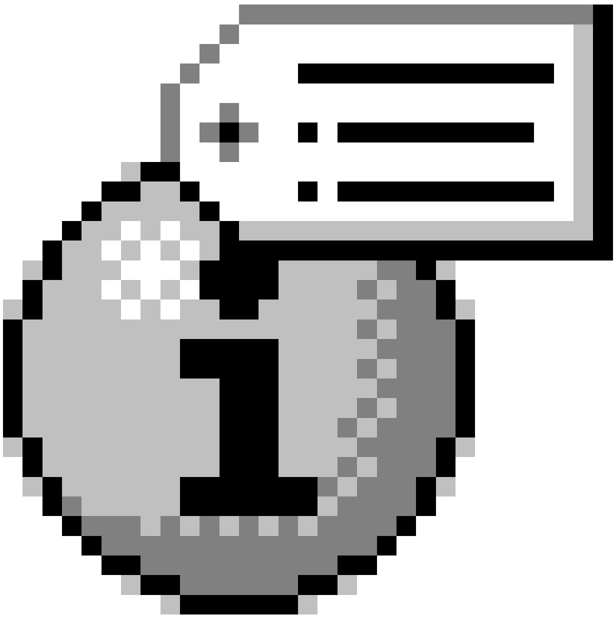
In the example diagram the package represents the information action types described by the ENVRI RM. The stereotype for the package is invariant schemata, which indicates that these are the parts of the model that are stable. The RM also provides a detailed description of each action in text.

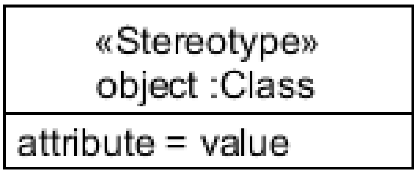
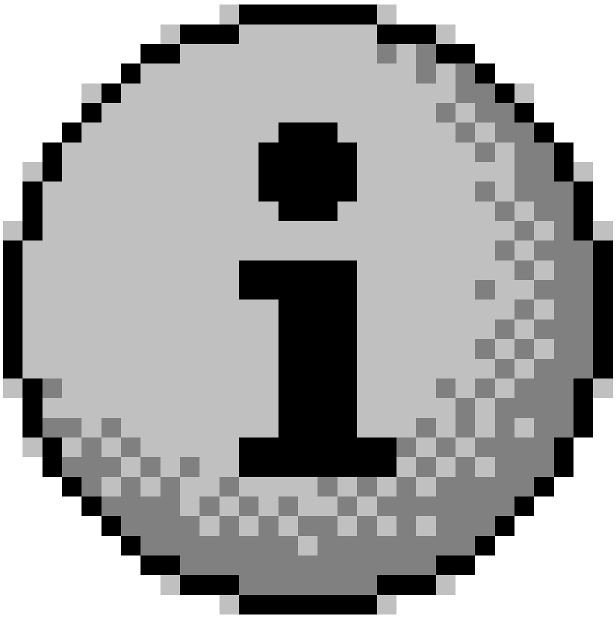
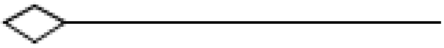
Information Object Instances

IV Objects instances are represented using an object diagram. The type of diagram is similar to the class diagram with the difference that the entities represented are objects not classes. Object instances have a specific state and this can change depending on the moment when the object is observed. Object instances are useful for representing the dynamic nature of the systems.

Table 6 Notation for information object instances diagrams

Figure	Description

 	A package, in UML notation, is a grouping element. Package is used "to group elements, and to provide a namespace for the grouped elements". A package may contain other packages, thus providing for a hierarchical organization of packages. Classes, objects, use cases, components, nodes, node instances etc. can all be organized as packages, enabling a manageable organization of the elements of UML models.
 	Classes are used to represent information objects in the RM. Classes can have additional compartments to express properties (called attributes) and behaviours (called methods). Leaving the compartments blank means that the behaviour and attributes are undefined at the time of creating the diagram. Name tag indicates the name of the class. Classes are named using no spaces and capitalising the first letter of each word that makes up the name, camelcase. The stereotype indicates the namespace where the class is grouped. Sometimes the stereotype can be an image. The image can be used in place of the figure. For ODP, the stereotype for information object is an “i” icon with a tag on top: 

	Objects are used to represent object instances in the RM. Name tag indicates of the entity The set of attributes with a value assigned characterises the state of the object. The stereotype indicates the namespace where the object is grouped. Sometimes the stereotype can be an image. The image can be used in place of the figure. For ODP, the stereotype for information object instance is an “i” icon: 
	Aggregation indicates an association that represents a part-whole or part-of relationship. Aggregation is represented with an arc with a blank rhombus decoration. The end with the blank rhombus indicates the composite and the other connects to the component.

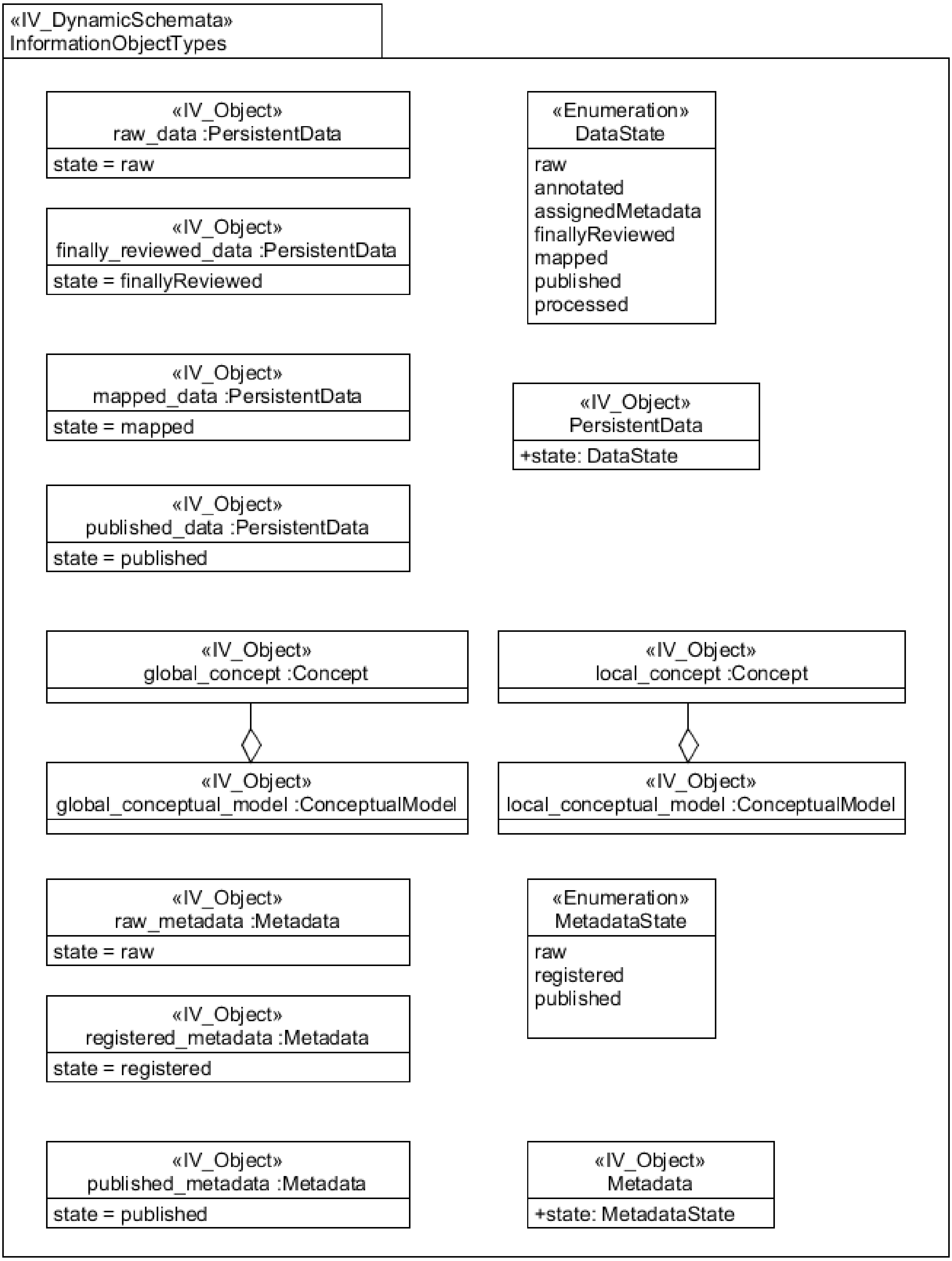


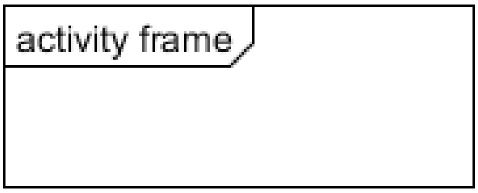
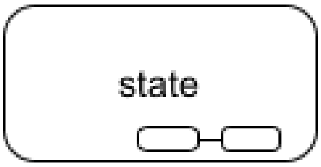
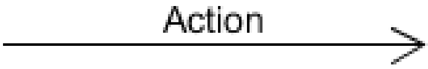
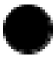
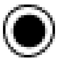
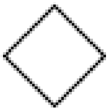
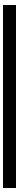
Figure 6 Example of an IV Object diagram

In the example diagram the package represents a collection of some information object instances. The stereotype for the package is dynamic schemata, which indicates that these are the parts of the model that can change depending on when the system is observed. The diagram presents four sample instances of persistent data objects and three examples of metadata objects. The diagram also includes the class definitions of persistent data and metadata objects for reference

State Diagrams

IV Object instances can have different states during their lifespan. The basic information objects persistent data and metadata have specific sets of states associated to them. The state changes, together with the IV Activities can be used to model the behaviour of data as it is managed by the RI. For this we use a state machine diagram. The main components of state machine diagrams are activity frames, states, activities, and pseudo-states

Table 7 Notation for information object instances diagrams

Figure	Description
	Frames are used to indicate the information object instance being represented. The name indicates the information object instance being modelled
	States are used to represent the state of an information object resulting from the effects of an IV action The name tag indicates the state reached by the information object The small decoration in the box can be included to indicate that the state is complex and can be subdivided into sub-states
	The arcs connecting states represent information actions applied to objects at a given state. The arrow end indicates the resulting state, the undecorated end indicates the initial state
	A filled circle is a pseudo-state, it can be used to model a start state or an intermediate connecting state
	A circle with a smaller filled circle in the middle is a pseudo -state to model an end state
	Decision pseudo-state, is used to model an exclusive fork in the execution of activities. It can also be used to model exclusive joins after forks.
	Fork/merge pseudo-state, is used to model a forks and joints in the execution of activities.

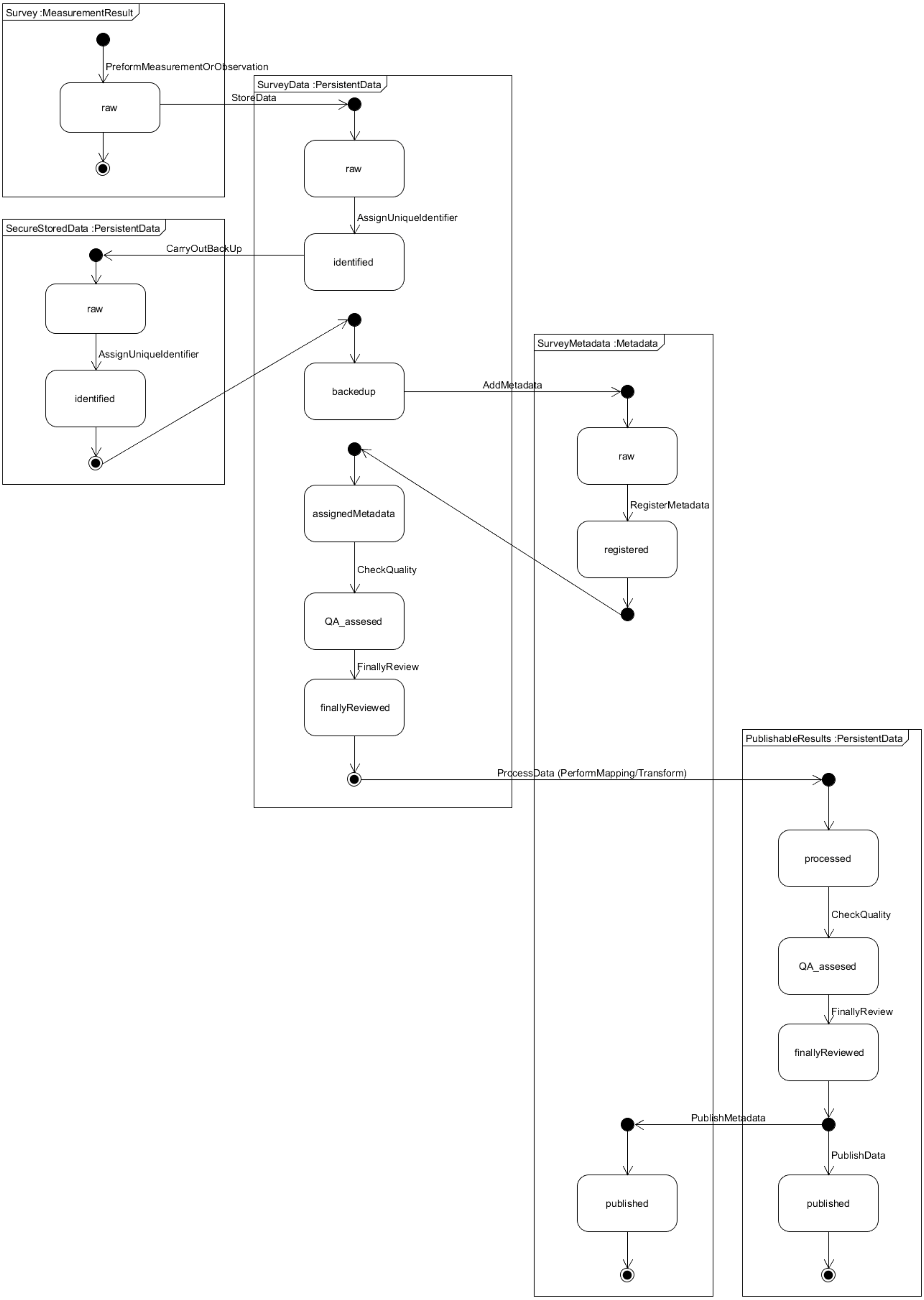


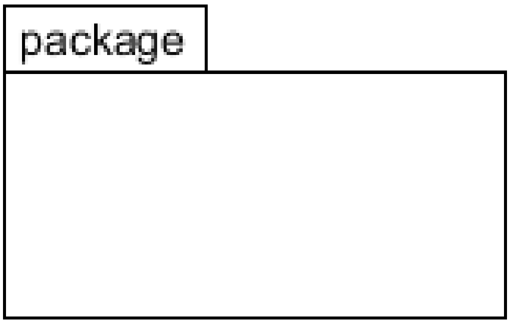
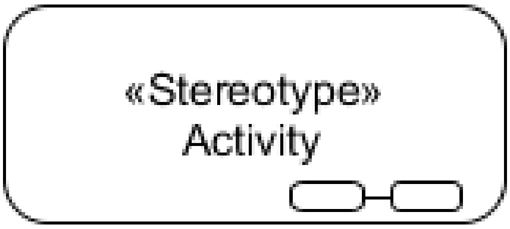
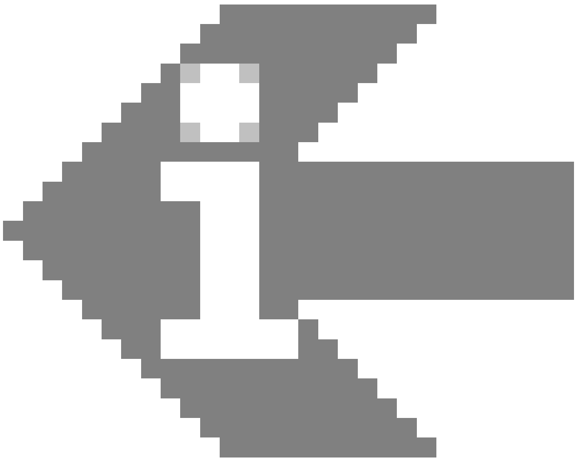
Figure 7 Example of an IV Information Object Evolution diagram

In the example diagram, five information object instances are presented. The possible transitions between states are indicated with arcs labelled using the names of IV actions.

Evolution of information objects

The evolution of information objects can also be represented using activity diagrams. Activity diagrams combine IV Object Instances and IV actions can also be combined into

Table 8 Notation for information object evolution with activity diagrams

Figure	Description
 	A package, in UML notation, is a grouping element. Package is used "to group elements, and to provide a namespace for the grouped elements". A package may contain other packages, thus providing for a hierarchical organization of packages. Classes, objects, use cases, components, nodes, node instances etc. can all be organized as packages, enabling a manageable organization of the elements of UML models.
 	Activities are used to represent action in the RM. Name tag indicates the name of the action. Actions are named using a short phrase that describes the event or action being represented. The small decoration in the box indicates that the action is complex and can be subdivided into smaller tasks. A stereotype can be used to indicate the namespace where the activity is grouped. Sometimes the stereotype can be an image. The stereotype image can be used in place of the figure. For ODP, the stereotype for information action is an arrow icon with a lowercase “i”: 

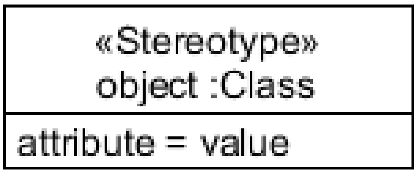
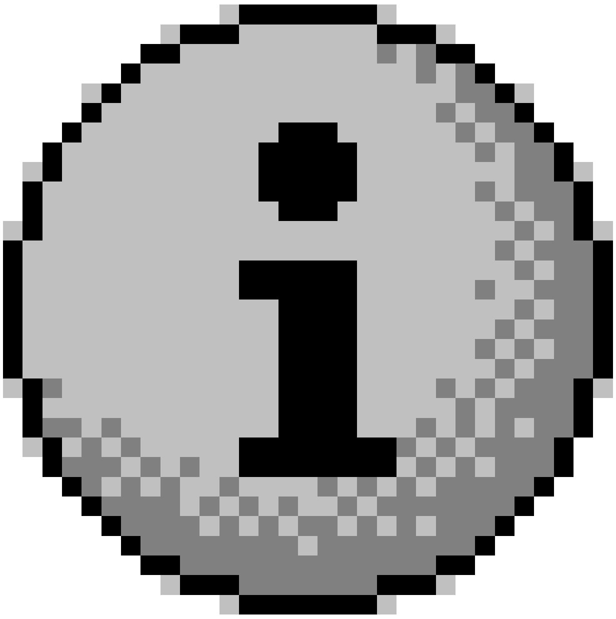
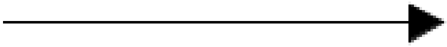
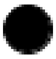
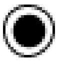
	Objects are used to represent object instances in the RM. Name tag indicates of the entity The set of attributes with a value assigned characterises the state of the object. The stereotype indicates the namespace where the object is grouped. Sometimes the stereotype can be an image. The image can be used in place of the figure. For ODP, the stereotype for information object instance is an “i” icon: 
	The arcs connecting states represent transitions between information actions. Arcs can connect activities to information object instances, indicating the result of performing an action. When linking an object to and action, the arc indicates that the object is part of the input used to perform that action.
	A filled circle is used to model the start of a set of actions
	A circle with a smaller filled circle in the middle is used to model an end state

Figure 8 Example of an IV Information Object Evolution using an activity diagram

In the example diagram, an overview of the evolution of data in a RI is presented

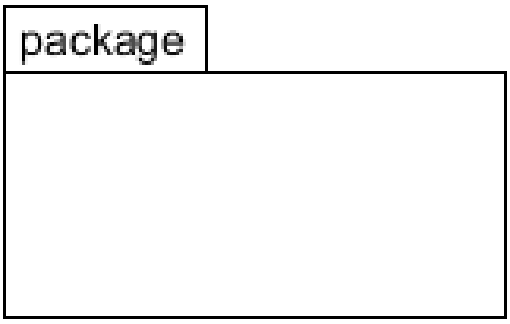
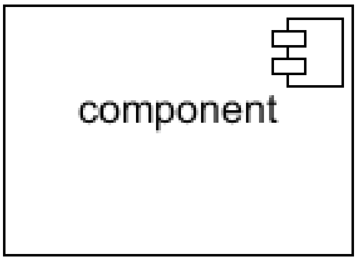
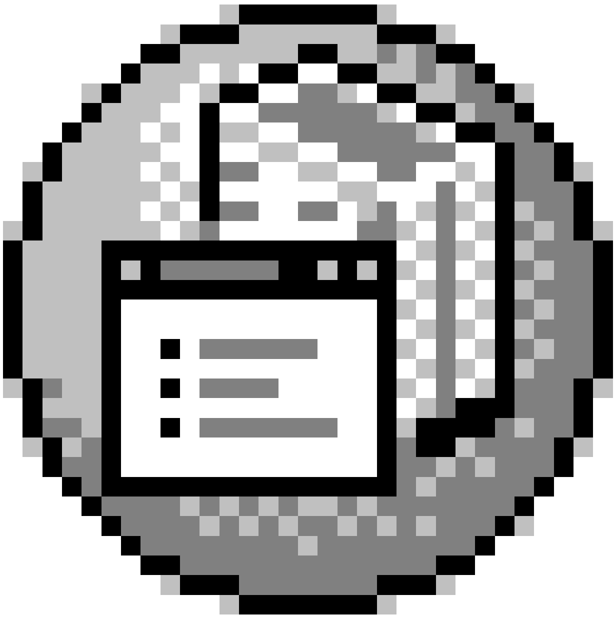
Notation of Computational Viewpoint Models

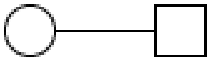
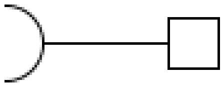
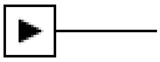
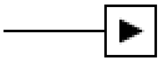
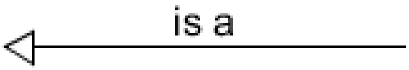
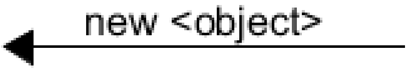
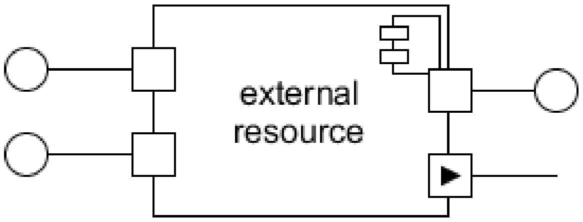
Computational Objects

In the ENVRI RM, component diagrams are used for the representation of computational objects and interfaces.

Table 9 Notation for information object instances diagrams

Figure	Description

	A package, in UML notation, is a grouping element. Package is used "to group elements, and to provide a namespace for the grouped elements". A package may contain other packages, thus providing for a hierarchical organization of packages. Classes, objects, use cases, components, nodes, node instances etc. can all be organized as packages, enabling a manageable organization of the elements of UML models.
	Components are used to represent computational objects. The box contains the name of the computational objet and a decoration indicating that it is a component (UML standard). Components can also have a stereotype, and an image associated with that stereotype. In ODP the stereotype image for computational objects is the icon of a box with a class tag in front of it: 

 server interface and port  client interface and port  Streaming interface producer  Streaming interface consumer	Ports and interfaces are used to represent the means of communication between objects. A small box in the border of an object is used to represent a port. A blank circle connected by an arc to a port represents a server interface A semicircle with an arc connected to a port represents a client interface A port with an arc and an arrow pointing away from the object represents a producer streaming interface A port with an arc and an arrow pointing towards the object represents a consumer streaming interface
 	Generalisation is used to indicate if one object extends another, this can be illustrated using an unfilled arrow from the sub-object to the parent, with the annotation 'is a'.
 	The ability to create objects is noted by a single filled arrow extending from the creating object to the object being created, with the annotation 'new <object>'.
 	Example of a computational object with four ports and four interfaces

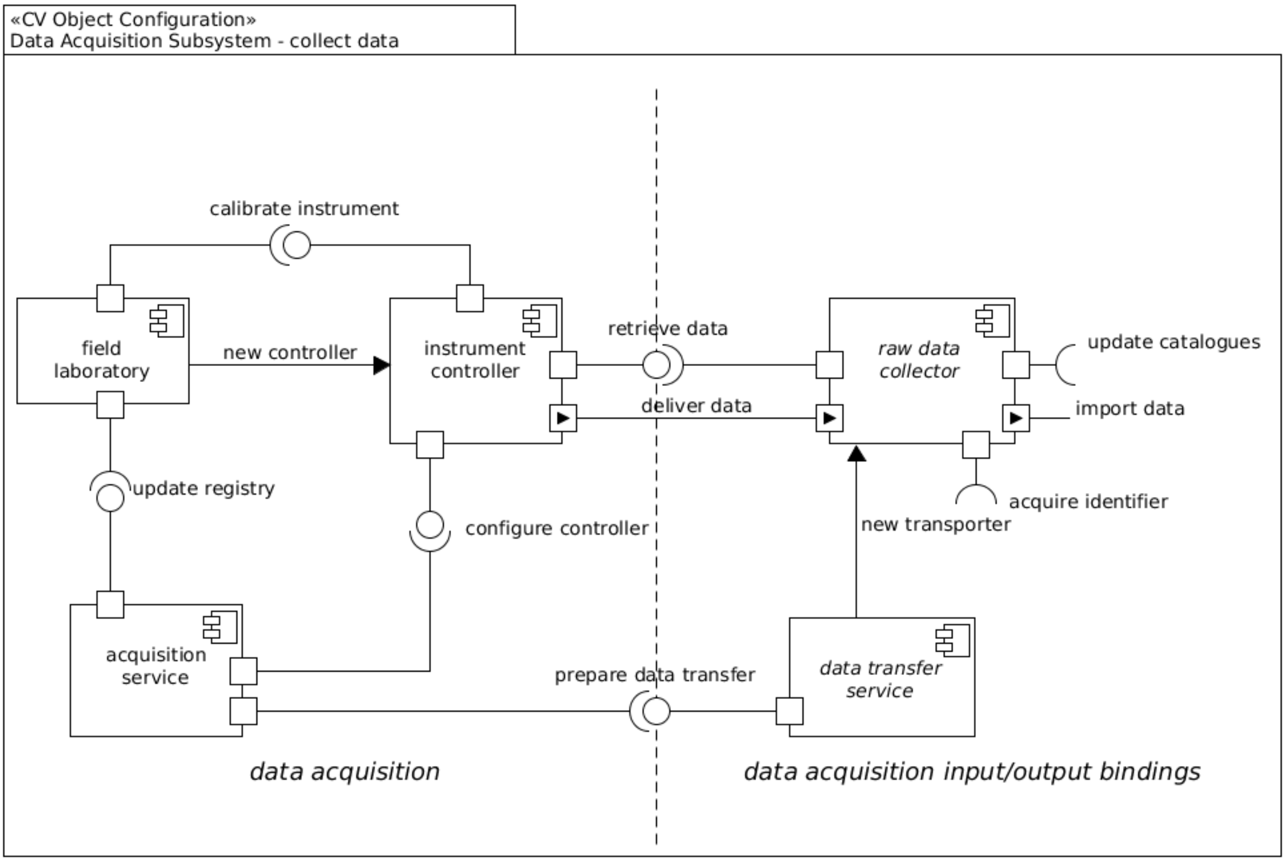
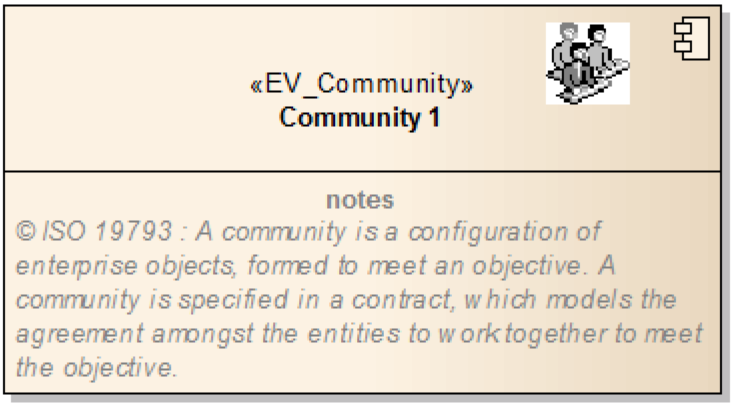


Figure 9 Example of the CV Objects for data acquisition

In the example diagram, three computational objects are presented. Balls and sockets are matched and the names of the client/server interfaces are supposed to be the same. In the example, the field laboratory client interface “calibrate instrument” is connected to the instrument controller server interface “calibrate instrument”

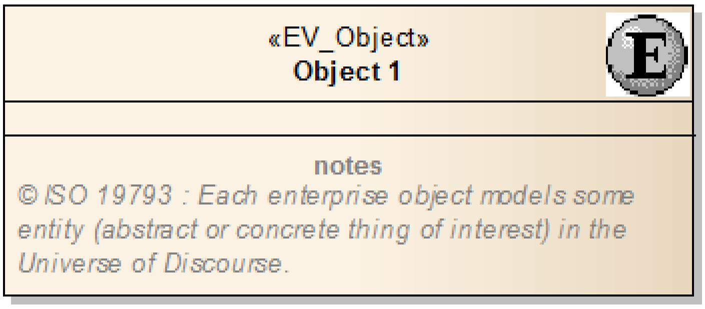
UML4ODP Graphical Notation

Enterprise viewpoint


«EV_Community»
Community 1

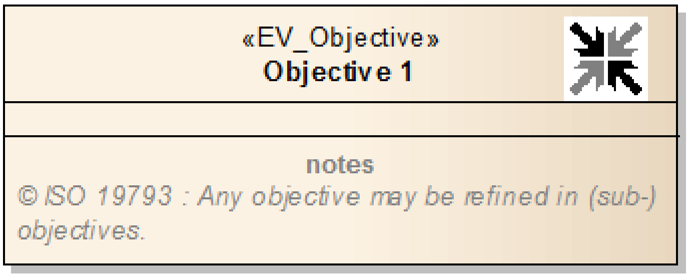
notes

© ISO 19793 : A community is a configuration of enterprise objects, formed to meet an objective. A community is specified in a contract, which models the agreement amongst the entities to work together to meet the objective.


«EV_Object»
Object 1

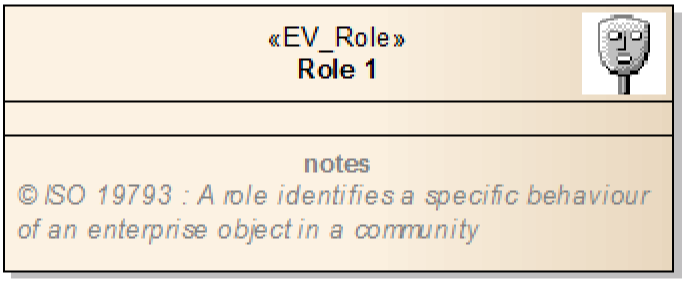
notes

© ISO 19793 : Each enterprise object models some entity (abstract or concrete thing of interest) in the Universe of Discourse.


«EV_Objective»
Objective 1

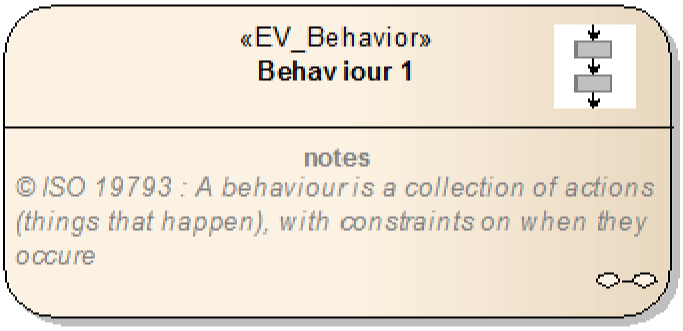
notes

© ISO 19793 : Any objective may be refined in (sub-) objectives.


«EV_Role»
Role 1

notes

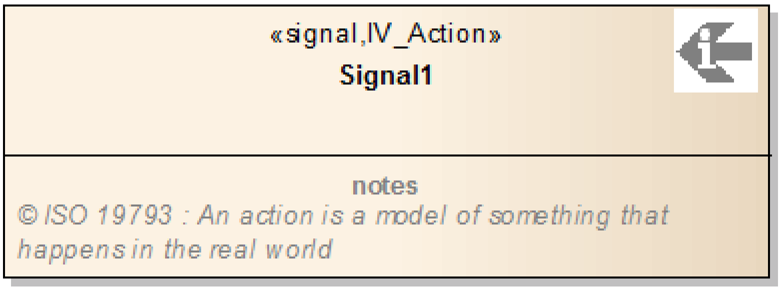
© ISO 19793 : A role identifies a specific behaviour of an enterprise object in a community


«EV_Behavior»
Behaviour 1

notes

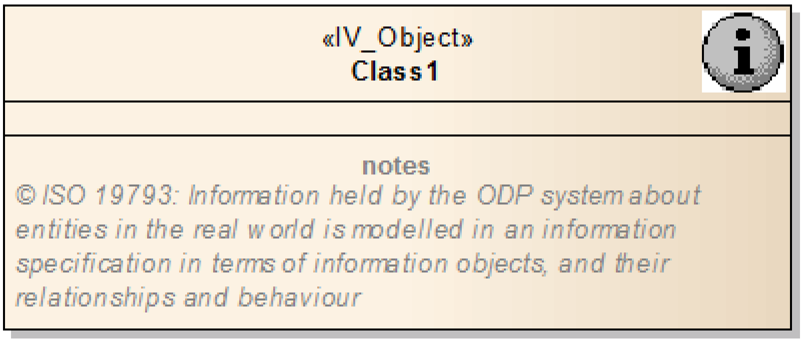
© ISO 19793 : A behaviour is a collection of actions (things that happen), with constraints on when they occure

Information viewpoint



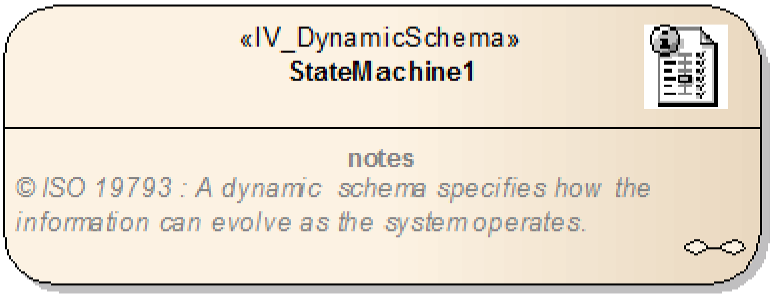
«signal,IV_Action»
Signal1

notes
© ISO 19793 : An action is a model of something that happens in the real world



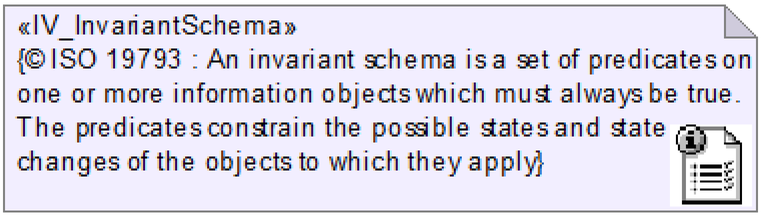
«IV_Object»
Class1

notes
© ISO 19793: Information held by the ODP system about entities in the real world is modelled in an information specification in terms of information objects, and their relationships and behaviour

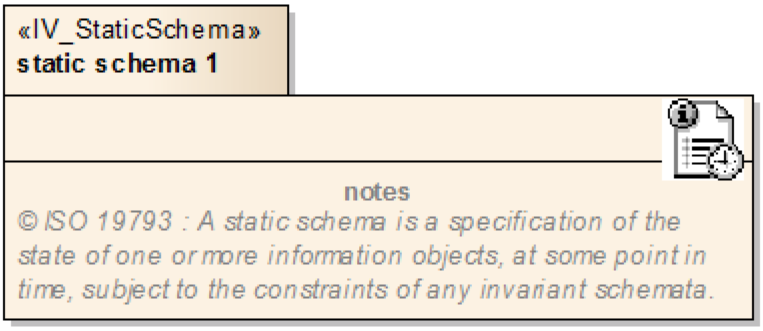


«IV_DynamicSchema»
StateMachine1

notes
© ISO 19793 : A dynamic schema specifies how the information can evolve as the system operates.



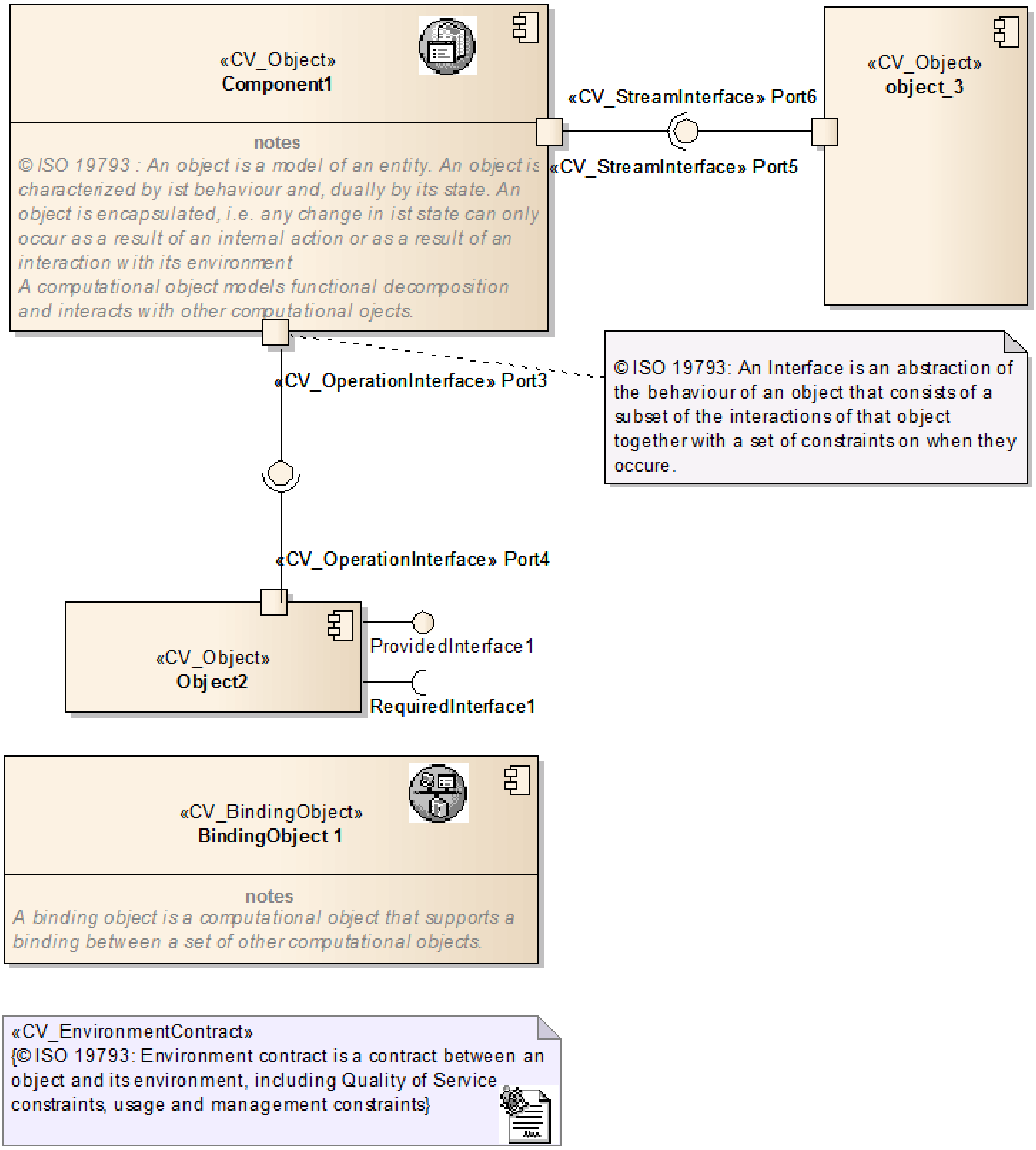
«IV_InvariantSchema»
{© ISO 19793 : An invariant schema is a set of predicates on one or more information objects which must always be true. The predicates constrain the possible states and state changes of the objects to which they apply}



«IV_StaticSchema»
static schema 1

notes
© ISO 19793 : A static schema is a specification of the state of one or more information objects, at some point in time, subject to the constraints of any invariant schemata.

computational viewpoint


«CV_Object»
Component1
notes
© ISO 19793 : An object is a model of an entity. An object is characterized by ist behaviour and, dually by its state. An object is encapsulated, i.e. any change in ist state can only occur as a result of an internal action or as a result of an interaction with its environment
A computational object models functional decomposition and interacts with other computational ojects.
«CV_StreamInterface» Port6
«CV_StreamInterface» Port5
«CV_Object»
object_3
«CV_OperationInterface» Port3
© ISO 19793: An Interface is an abstraction of the behaviour of an object that consists of a subset of the interactions of that object together with a set of constraints on when they occure.
«CV_OperationInterface» Port4
«CV_Object»
Object2
ProvidedInterface1
RequiredInterface1
«CV_BindingObject»
BindingObject 1
notes
A binding object is a computational object that supports a binding between a set of other computational objects.
«CV_EnvironmentContract»
{© ISO 19793: Environment contract is a contract between an object and its environment, including Quality of Service constraints, usage and management constraints}

Appendix D Guidlines for using the Reference Model

Introduction

The development of the ENVRI Reference Model provides the ESFRI Environmental Research Infrastructures with a common ontological framework for description and characterisation of computational and storage infrastructures, and provides them a community standard to help achieve greater levels of interoperability between their heterogeneous resources.

The Reference Model defines a conceptual model that captures computational requirements and state-of-the-art design experiences. In a sense, the model reveals a snapshot of the existing landscape of the ESFRI environmental science research infrastructures at a high level of abstraction.

In order to help Reference Model users map the abstraction to concretions, so as to better apply the knowledge in their daily practices, we prepare this guideline that introduces our own experiments with the Reference Model, and in doing so reveal the principles of usage. These principles are neither bound nor enforced. They are not mandatory for users to follow. The intention is to provide users with a way of thinking, which may lead to exploration of the model itself and inspire the discovery of various way of using the model.

Rather than going through each model term and explaining the meaning of it, we use a set of practical examples, each of them illustrating some aspects of the usage of the reference model as well as introducing a number of model concepts.

Initially, examples are selected with the aim to serve **Intended Audience** within the community of ESFRI Environmental Research Infrastructures. We use scenarios that are familiar to our users, and include information that may be of interest to the community and perhaps benefit their work.

To collect these examples, we used a template with 5 questions:

1. What is this use case about? *Describe the purpose of the use case, and any background information.*
2. How can the reference model be used in this use case?
3. What are the results of using the reference model? *Evidence of usefulness/utility.*
4. What are the benefits of using the reference model? *Demonstrate specific cases of things that could not have been achieved without the RM.*
5. Are there any problems with using the reference model in this use case? *Feedback from users.*

These questions proved to be helpful in organising investigation activities. We encourage readers also to use this template to structure newly developed stories and share them with us so as to inspire others.

With limited resources, only few examples are included; these will be extended when more resources are available for future investigations.

How to Use the Guideline

A collection of examples demonstrating usage of the ENVRI Reference Model is given below. Different examples may serve different purposes. Some of them merely illustrate a different way of using the reference model (e.g., Example 5), while others also intend to introduce model concepts where many terms are highlighted with clickable links. Please click those highlighted concepts that will re-locate you to the related definitions and specifications in the Reference Model. Be sure to go through all terms marked with 💡 -- some of them, though repeated, will guide you to a different part of the model. By visiting all linked contents, you will have explored 90% of the most important model content. (Note, terms marked with 💡 are also model concepts which link to content you might have visited before.)

Examples of Using the Reference Model

Example 1: Using the Reference Model to Guide Research Activities (EISCAT 3D - EGI)

Example 2: Using the Reference Model as an Analysis Tool (EUDAT)

Example 3: Using the Reference Model in documentation (EMSO)

Example 4: Using the Reference Model as design reference (EPOS)

Example 5: Using the Reference Model to drive implementations of common services (WP4 practices)

Conclusions

Using a number of examples, we have shown that by using the Reference Model, a ESFRI ENV RI could benefit from:

- **A set of ready-to-use terminology with a publicly-accessible reference base**, which can be used to describe requirements and architectural features of an infrastructure, and serve as a common language in communication materials; in particular, with an external community without any specific knowledge of the scientific domain being addressed.
- **A uniform framework with well-defined subsystems** of components specified from different complementary viewpoints (Science, Information and Computation), which promotes structural thinking in constructions of system architectures, and can be used as a research tool for comparison and analysis of heterogeneous infrastructures.
- **A knowledge base capturing existing requirements and state-of-the-art design experiences**. The information provided can be referred to in various system analysis tasks, to guide design and implementation activities, and to drive the development of common services.

When future resources become available, we will conduct more investigations, including:

- We will assist our users to get hand on the Reference Model and exploit new ways of using it.
- We will assist the development of the common services.
- We will use the Reference Model to bridge ESFRI ENV RIs with external communities (such as, RDA), projects (such as, GEOSS, DataOne, EUDAT and EGI), and standards (such as, INSPIRE, OGC, and the Digital Library Reference Model). These will provide ESFRI ENV RIs an overview of related technologies, and possible solutions for the integrations.
- We also have a plan to experiment with the Reference Model as a guide to train the next generation data scientists.

Tutorials

- ENVRI Reference Model: an Overview. [**.pptx**]
- Main Processes of the ENVRI Reference Model – Corresponding Viewpoint [**.pptx**]

Example 1: Using the Reference Model to Guide Research Activities (EISCAT 3D - EGI)

Descriptions of the Example

This example explains the usage of the Reference Model in a pilot project that investigates the big data strategies for the EISCAT 3D research infrastructure. The Reference Model serves as a knowledge base to guide various research activities.

EISCAT, the *E*uropean *I*ncoherent *Scat*ter Scientific Association, was established to conduct research on the lower, middle and upper atmosphere and ionosphere using the incoherent scatter radar technique. This technique is the most powerful ground-based tool for these research applications. A next generation incoherent scatter radar system, EISCAT 3D, is being designed. The multi-static radars to be used will be a tool to carry out plasma physics experiments in the natural environment, a novel atmospheric monitoring instrument for climate and space weather studies, and an essential element in multi-instrument campaigns to study the polar ionosphere and magnetosphere. It will be a world-leading international research infrastructure, using the incoherent scatter technique to study how the Earth's atmosphere is coupled to space.

The design of the EISCAT 3D opens up opportunities for physicists to explore many new research fields. On the other hand, it also introduces significant challenges in handling large-scale experimental data that will be massively generated at great speeds and volumes. During its first operation stage in 2018, EISCAT 3D will produce 5PB data per year, and the total data volume will rise up to 40PB per year in its full operations stage in 2023. This challenge is typically referred to as a big data problem and requires solutions beyond the capabilities of conventional database technologies.

EISCAT is currently considering the use of e-Science technologies to deliver strategies for handling its big data products. Advanced e-Science infrastructure projects such as EGI, PRACE, and their enabling technologies are making large-scale computational capacities more accessible to researchers of all scientific disciplines. Emerging infrastructures, such as cloud systems proposed by the Helix Nebula project and by the EGI Federated Cloud Task Force, or the data infrastructure to be provided by EUDAT will extend possibilities even further.

As a potential of e-science partner for EISCAT, we present EGI. EGI was established in 2010 as a Europe-wide federation of national computing and storage resources. The EGI collaboration is coordinated by EGI.eu, a not-for-profit foundation created to manage the infrastructure on behalf of its participants: National Grid Initiatives and European Intergovernmental Research Organisations. Resources in EGI are provided by about 350 resource centres from the NGIs who are distributed across 55 countries in Europe, the Asia-Pacific region, Canada and Latin America. These providers operate more than 370,000 logical CPUs, 248 PB disk and 176 PB of disk capacity (June 2013 statistics) to drive research and innovation in Europe and beyond.

Since February 2013, a pilot project has been set up within ENVRI, which establishes a partnership between EISCAT, EGI and EUDAT, aiming to identify and allocate solutions that directly benefit EISCAT 3D, which can also be reused in other ESFRI projects involved in ENVRI. ENVRI WP3 has been involved in this investigation, and uses the Reference Model to guide various research activities, including;

- Analysis of the EISCAT 3D data infrastructure; Capturing requirements from the EISCAT 3D scientific community concerning applications that work with and process data.
- Analysis of EGI and EUDAT services; Identifying the gaps between the generic service infrastructures of these providers and the domain-specific requirements of EISCAT 3D.
- Provide recommendations to EISCAT 3D for the setup of up a big data strategy and a big data infrastructure for its community. Setup demonstrators/proof-of-concept systems if resources permit.

Having fulfilled these tasks, the Reference Model is proving to be useful as a knowledge base that can be referred when conducting various system analysis and design activities.

How to Use the Reference Model

In the following, we describe how the Reference Model is used to conduct several system analysis tasks.

Analysis of the EISCAT 3D Data Infrastructure

The initial challenge for the pilot project is to understand the EISCAT 3D data infrastructure. The existing design documents of EISCAT 3D has been focused on the incoherent scatter radar technologies. As shown in Figure 1, its data infrastructure is embedded within the overall design of the observatory system that is difficult for a computer scientist/technologist having little physics knowledge background to understand.

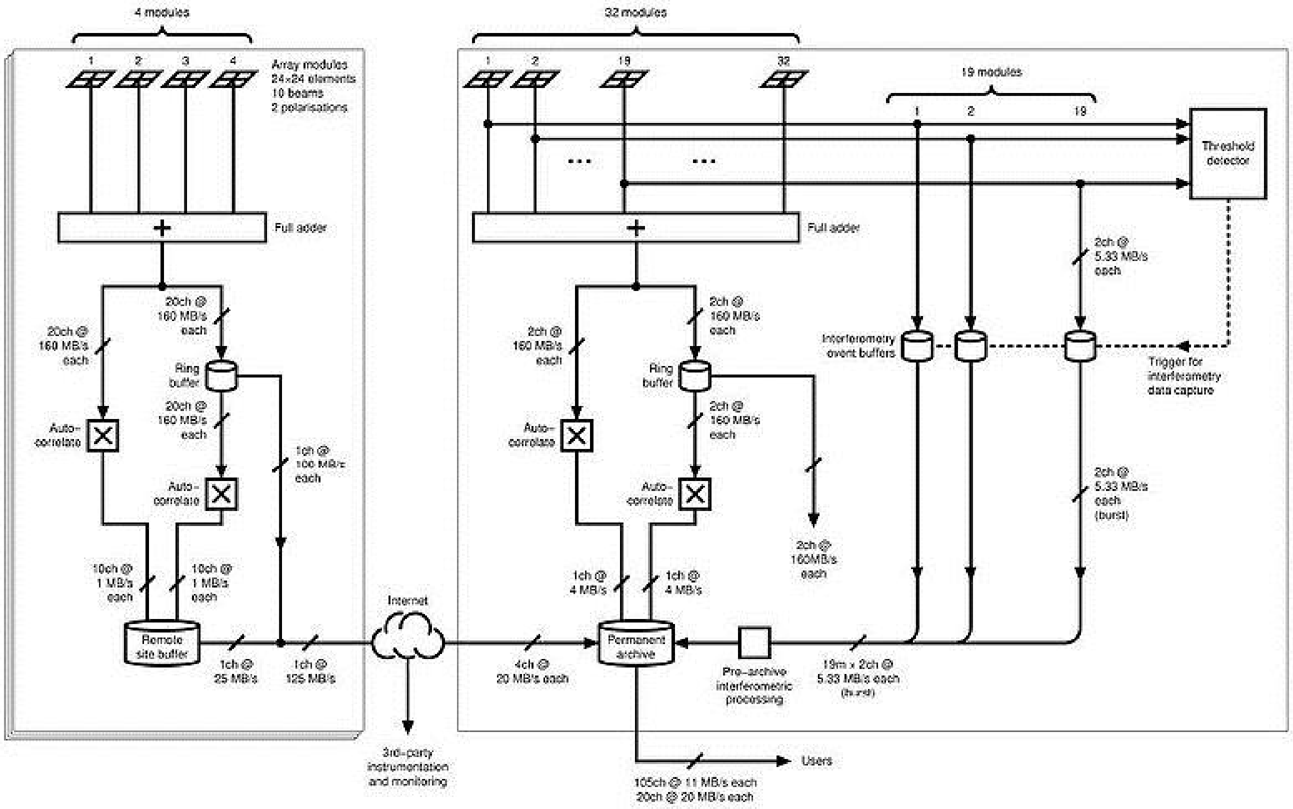


Figure 1: The original design of EISCAT 3D data infrastructure is embedded within the overall observatory system design

We use the 💡 **ENVRI_Common_Subsystem** framework to decompose the computational elements, clarifying the boundary between the radar network and data infrastructure, which results in Figure 2. This diagram now, instead of Figure 1, is frequently used in presentations and discussions of the EISCAT 3D data infrastructure.

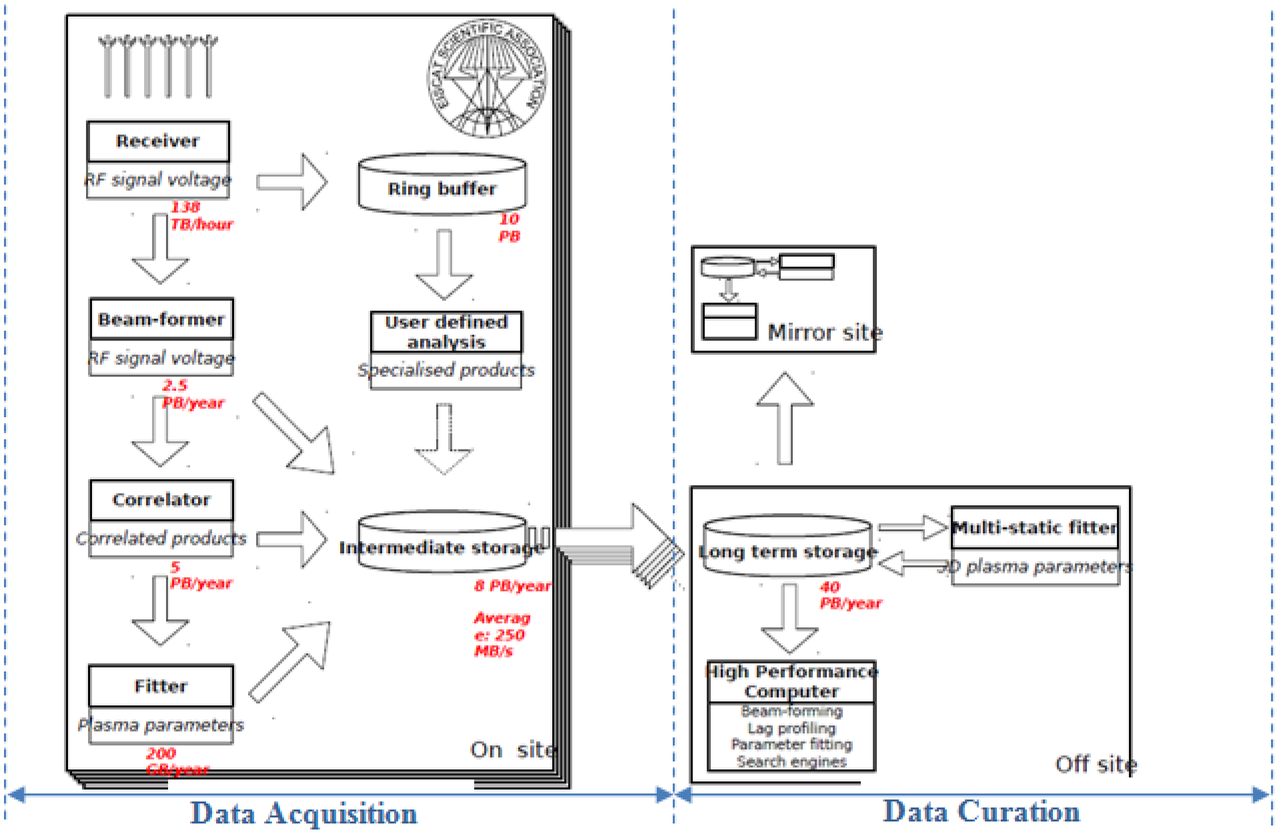


Figure 2: Using the 5 ENVRI Common Subsystem to interpret the EISCAT 3D data infrastructure makes it easy to communicate with computer scientists/technologists

Figure 2 illustrates that the EISCAT 3D functional components can be placed into 2 ENVRI common subsystems, **subsys_acq** and **subsys_cur**. Briefly, at the **subsys_acq**, the raw signal voltage data will be generated by the antenna *Receivers* at the speed of 125 TB/hr, and be temporarily stored in a *Ring buffer*. A second stream of RF signal voltages will be passed to a *Beam-former* to generate the beam-formed data (1MHz). Continually, the beam-formed data will be processed by a *Correlator* to generate correlation analysis data based on standard methods. Then, the correlation data will be delivered to a *Fitter* to produce the fitted data (1GB/year). In order to support different user requirements, EISCAT 3D will allow users to access and process the raw voltage data in the *Ring buffer* and to generate the specialised products based on self-defined analysis algorithms. Both raw data and their products will be stored in *Intermediate storage* (11PB/year), from where they will be delivered to the central site within the curation subsystem.

In **the curation subsystem** , *Long-Term Storage* will preserve the raw voltage data and their products. A *High Performance Computer* will be used for data searching and processing (e.g., beam forming, lag profiling or other correlation, and parameter fitting). Searching facilities will enable user to search over all data products and to identify significant data signatures. A *Multi-static fitter* will be installed to process the stored raw voltage data to generate the 3D plasma parameters that will then be stored back in *Long-Term Storage*. A complete copy of *Long-Term Storage* data will be established at mirror sites; related data processing and searching tools will be provided.

While it is made clear that the design specification covers 2 of 5 common subsystems described in the ENVRI Reference Model, we understand functionalities of the other 3 subsystems are currently missing. The reason of this is likely due to resource limitations. However, the absent 3 subsystems are crucial for a big data system such as EISCAT 3D. Without providing services to support data discovery, access, processing and user community, the value of EISCAT 3D big data cannot be unlocked, and expensively generated and archived scientific data will be useless.

Using the Reference Model as the analysis tool, we identified the missing pieces of the design specification, which gives the direction for future investigation.

Analysis of EGI Enabling Services and Construction of an Integrated Infrastructure

We need to understand the functionalities of EGI services and how to integrate them to support the EISCAT 3D requirements.

A set of generic services are enabled by the EGI e-Infrastructure, including:

- AMGA Metadata catalogue
- LFC File catalogue
- Storage elements
- File Transfer Service
- Portal for application development & hosting (e.g. SCI-BUS)
- Access control

Showing in Table 1, by examining the functionalities of the EGI services and mapping them to the ENVRI Reference Model computational model objects, we understand these services fall into 2 ENVRI common subsystems: Curation and Community Support.

Table 1: Mapping EGI Services to the Reference Model Elements (from **Computational Viewpoint** perspective)

EGI Services	ENVRI- RM Computational Objects	ENVRI Common Subsystem
AMGA Metadata catalogue	**catalogue service**	**CV Data Curation**
LFC File catalogue	**catalogue service**	**CV Data Curation**
Storage elements	**data store controller**	**CV Data Curation**
File Transfer Service	**data transfer service**	**CV Data Curation**
Portal for application development & hosting	**virtual laboratory**	**CV Data Use**
Access control	**security service**	**CV Data Use**

Above analysis gives clues to a solution for integrating the EGI technologies into the EISCAT 3D data infrastructure. Depicted in Figure 3, a secondary **CV Data Curation** (seen as the mirror site of the EISCAT 3D central archive in Figure 1) can be established using the EGI infrastructure and its services. Data from EISCAT 3D central archive (or the acquisition subsystem) can be staged into the EGI storages, and be managed using LFC File Catalogue and AMGA Metadata Catalogue. At the front end, an EISCAT science gateway can be established, seen as part of a **CV Data Use**, to provide access control (e.g., authentication, authorisation, and single sign-on) and application portals (e.g., to which processing and data- mining applications from EISCAT 3D can be plugged in).

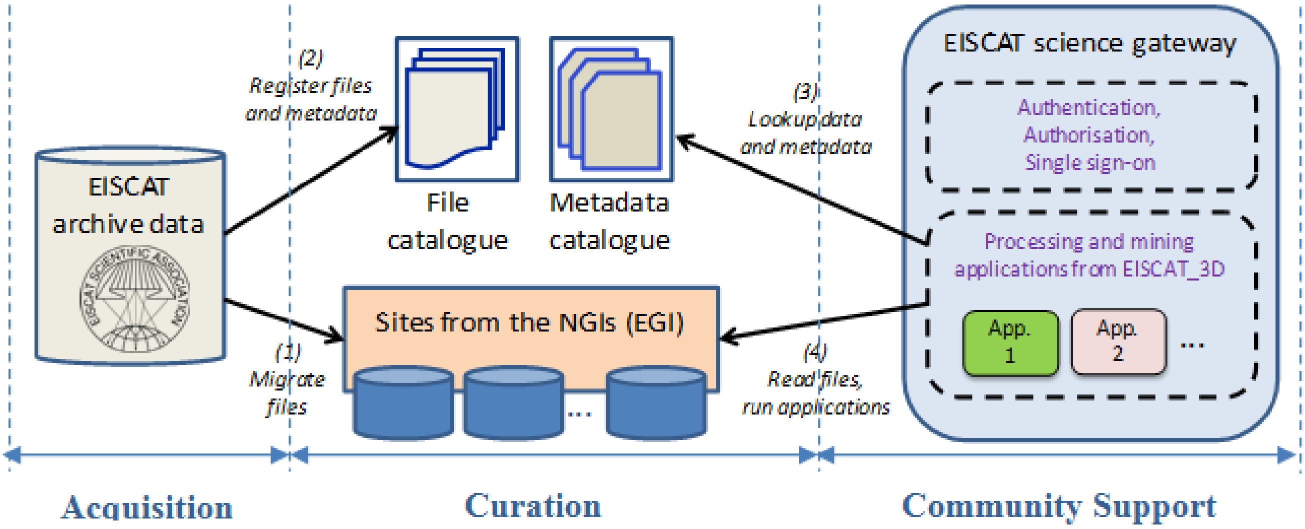

EISCAT archive data
(2) Register files and metadata
File catalogue
Metadata catalogue
(3) Look up data and metadata
EISCAT science gateway
Authentication, Authorisation, Single sign-on
Processing and mining applications from EISCAT_3D
App. 1
App. 2
...
Sites from the NGIs (EGI)
(1) Migrate files
(4) Read files, run applications
Acquisition
Curation
Community Support

Figure 3: An integrated infrastructure of EGI and EISCAT 3D

Using the Reference Model, functional elements of both EISCAT 3D and EGI can be placed into a uniform framework, which provides a way of thinking about the construction of the integrated infrastructure.

Evaluation of the Feasibilities of the EGI Infrastructures and Services in Supporting EISCAT 3D Requirements

Using the common framework enabled by the Reference Model, we can analyse and compare the EGI and EUDAT generic service infrastructure and the requirements from a domain-specific data infrastructure such as EISCAT 3D, and we understand that there are significant gaps in-between, including but not limited to:

- Staging services to ship scientific data from observatory networks into the EGI generic service infrastructure (and to get the data off) are missing. Such a staging service should be able to transmit both big chunk of data (up to petabyte) and continuing updates/real-time data streams during operations. Such a service should satisfy performance requirements, including:
    - Robust. Environmental scientific research needs high quality data. In particularly, during important natural events, losing observation data is unaffordable. Fault-tolerance is desirable, which requests the transmission service can be self-recover from the interruption point without restarting the whole transmission process.
    - Fast, e.g., in the case of EISCAT 3D, the 10PB ring-buffer can only hold data for about 3 days, and the big observation data need to be transferred to the archive storage fast enough to avoid being overwritten.
    - Cheap, e.g., the observatory networks are remote from the EGI computing farm. Using high-capacity pipes are possible but expensive. Software solutions such as, intelligent network protocols, optimisation, data compression, are desirable.
- Cost effective large storage facilities and long-term archiving mechanisms are urgently needed. Environmental data, in particular for climate research, need to be preserved over the long-term to be useful. Being Grid-oriented, EGI is not designed for data archiving purposes. Although large storage capabilities are potentially available through NGI participants, EGI does not guarantee long-term persistent data preservation. Curation services such as advanced data identification, cataloguing and replication are absent from the EGI service list.
- The EGI infrastructure needs to adapt in order to handle emerging big- data phenomena. The challenge is how to integrate what is new with what already exists. Services such as job schedulers need to be redesigned to take into account the trade-off of moving big data; intelligent data partitioning services should be investigated as a way to improve the performance of big data processing.
- Advanced searching and data discovery facilities are urgently needed. It is often said that data volume, velocity, and variety define big data, but the unique characteristic of big data is the manner in which the value is discovered [38]. Unlike conventional analysis approaches where the simple summing of a known value reveals a result, big data analytics and the science behind them filter low value or low-density data to reveal high value or high-density data [38]. Novel approaches are needed to discover meaningful insights through deep, complex search, e.g., using machine learning, statistical modelling, graph algorithms. Without facilities to unlock the value of big data, expensively generated and archived scientific data will be useless.
- Community support services are insufficient. The big data phenomena will eventually lead to a new data-centric way of conceptualising, organising and carrying out research activities that could lead to an introduction of new approach to conducting science. A new generation of data scientists is emerging with new requirements. Service facilities should be planned to support their needs. These together should enable the EISCAT 3D community to design new applications that are capable to work with big data, and can implement these on cutting-edge European Distributed Computing Infrastructures.
- Currently, EUDAT has taken up the role to implement a collaborative data infrastructure, however only a few services are available, storage facilities are insufficient, and policies for usage are unclear. Among our current investigations, we are investigating the possibility of integrating EUDAT services into EGI infrastructure, seen as a layer on top of the EGI federated computing facility. The analysis of the EUDAT services is included in Example 2: Using the Reference Model as an Analysis Tool (EUDAT) of the Reference Model.

Summary

In this example, we have shown that the Reference Model could be used to conduct various system analysis tasks. Using the Reference Model we have:

- Clarified the boundary of EISCAT 3D data infrastructure and identified missing functionalities in the design;
- Provided a solution to integrate the EGI services into EISCAT 3D data infrastructure;

- Identified gaps between the EGI generic service infrastructure with the requirements from a domain specific research infrastructure, EISCAT 3D.

We have shown that the Reference Model offered a research infrastructure:

- A knowledge base containing useful information could be referred in various system analysis and design activities;
- A uniform platform into which computational elements of different infrastructures could be fitted, enabling comparison and analysis;
- A way of thinking of constructions of plausible system architectures.

Example 2: Using the Reference Model as an Analysis Tool (EUDAT)

Description of the Example

This study case provide an example for ESFRI Environmental Research Infrastructures project managers and architects to use the ENVRI Reference Model as an analysis tool to review an emerging technology, the EUDAT data infrastructure and its service components. Such an analysis can help them better understand the newly developed technologies and decide on how to make use of the generic services provided in their own research infrastructures.

The EU-funded EUDAT project is developing a pan-European data infrastructure supporting multiple research communities. Such a generic data infrastructure is seen as a layer in the overall European scientific e-infrastructure to complement the computing layer (EGI, DEISA, PRACE) and the networking layer (GEANT).

The design activities of EUDAT are driven by use-case-based community requirements EUDAT reviews the approaches and requirements of different communities, such as linguistics (CLARIN), solid earth sciences (EPOS), climate sciences (ENES), environmental sciences (LIFEWATCH), and biological and medical sciences (VPH), identifying common services, and provides computational solutions. Initially, 4 services are provided within EUDAT data infrastructure:

- **Safe replication**: which enables communities to replicate datasets -- using the integrated Rule-Oriented Data System (iRODS) as a replication middleware -- within data centre sites, with persistent identifiers automatically assigned to the digital objects in order to keep track of all the replicas;
- **Data staging**: which enables easy movement of large amounts of data between EUDAT storage resources and workspace areas on high-performance computing (HPC) systems to be further processed.
- **Metadata Catalogue**: which allows researchers to easily access metadata of data (or their collections) stored in EUDAT nodes. EUDAT will also harvest external metadata (which contains pointers to actual data) from stable metadata providers to create a comprehensive joint catalogue that will help researchers to find interesting data objects and collections.
- **Simple Storage**: which allows registered users to upload “long tail” data objects (large in number but small in size), and share such objects with other researchers.

We use the concepts developed in the ENVRI Reference Model to analyse the EUDAT data infrastructure and its service components. Only cursory analysis is provided, since the main purpose of the study case is to illustrate the usage of the ENVRI Reference model.

How to Use the Reference Model

Analysis of EUDAT common services and components

The ENVRI Reference Model models an archetypical environmental research infrastructure (RI). As a service infrastructure, EUDAT itself is therefore not an implementation of the Reference Model, but is rather a source of implementations for instances of objects required by any RI implementing the Model.

Table 1: Mapping EUDAT Services to the Reference Model Elements

EUDAT Services	Computational Viewpoint	ENVRI Common Subsystem
Safe replication	**data transfer service**	**CV Data Curation**
Staging	**data importer**	**CV Data Curation**
Metadata Catalogue	**catalogue service**	**CV Data Curation**
Simple Store	**data store controller**	**CV Data Curation**

From the ***computational*** perspective, EUDAT offers services that can be used to instantiate various objects in the Reference Model. For example EUDAT’s Safe Replication facilities can implement various required services within the **subsys_cur** of an environmental RI:

- **CV Data Acquisition**: EUDAT does not offer facilities for **subsys_acq**, relying on data already gathered by client RIs.
- **CV Data Curation**: EUDAT can provide instances of any of the computational objects used for data curation (including **data store controller**, **data transfer service** and **catalogue service**), either in place of or complementary to instances provided by an environmental RI – the extent to which EUDAT assumes the curation role for an infrastructure will vary from case-to-case.
- **CV Data Access**: Data access to EUDAT curated data is brokered by EUDAT, whilst the RI would broker RI-curated data. In practice the RI **broker** would sit in front of the EUDAT broker, forwarding data requests that involve data delegated to EUDAT.
- **CV Data Processing**: EUDAT do not offer data processing (beyond *metadata annotation*) as a core service; *workflow*

enactment is being investigated as a future service however, which would allow a later version of the EUDAT platform to implement elements of a **subsys_pro**.

- Whilst certain aspects of EUDAT such as the Simple Store for researchers might be directly accessible as an independent **gate way service** , in general EUDAT sits behind a client RI, its services hidden behind the RI's native services from the perspective of the RI's user community. It would be likely however that the 'virtual laboratories' by which community groups interact with an RI would be in some way augmented by EUDAT services; in particular, implementations of the **security service** would integrate the EUDAT AAI service to allow seamless integration of EUDAT-held datasets with locally-held RI datasets.

The most immediately apparent conclusion that can be drawn from cursory analysis of EUDAT services in the context of the Reference Model is that EUDAT can potentially implement the entire **CV Data Curation** of an environmental RI; however in practice, one would expect that an RI would retain a certain amount of data locally (particularly raw data that is expensive to transfer off-site), necessitating a more nuanced division of labour between the RI and EUDAT. In particular, EUDAT provides replication services, allowing the co-existence of RI and EUDAT data stores holding the same data, and EUDAT provides metadata (including global persistent identifier) services, allowing EUDAT to provide any **catalogue service** (probably complementary to catalogue services maintained by an environmental RI itself). The delegation of services will be a product of negotiation between the environmental RI and the EUDAT project (some degree of automation may be possible, but likely sufficient for only smaller projects).

Summary

The principal potential benefit of using the Reference Model in general is the ability to precisely identify components required by an environmental RI and then identify how (if at all) the RI implements those components. In the EUDAT context, EUDAT provides a number of services that implement certain components (primarily in **CV Data Curation**); it should therefore be possible to identify the equivalent services in a modelled RI and determine whether or not there is a benefit to delegating those services to EUDAT. This decision may be based on cost (particularly related to economies of scale) and development time (in cases where the RI has not yet implemented the service, but may be able to use the EUDAT service instead).

Example 3: Using the Reference Model in documentation (EMSO)

Descriptions of the Example

Researchers and architects of an ESFRI Environmental Research Infrastructure often encounter requests to describe their infrastructure, to introduce its particular architectural features, or to explain system requirements. The Reference Model offers a set of ready-to-use terminology with explicit definitions, which can be applied to various documentations. This example tells how the Reference Model has been used as a common language in writings to communicate with a community other than environmental science.

The Research Data Alliance (RDA) is established to accelerate international data-driven innovation and discovery by facilitating research data sharing and exchange, use and re-use, standards harmonization, and discoverability. This will be achieved through the development and adoption of infrastructure, policy, practice, standards, and other deliverables.

ENVRI has been actively supporting the RDA activities and made various contributions. In particular, ENVRI has been accepted as one of the use cases by the RDA Data Foundation and Terminology (DFT) working group, which has been set up to gather emerging requirements as well as to test research outcomes.

In preparing the use case, researchers and architects from two ENVRI-participating research infrastructures, EMSO and EPOS, used the terms and concepts defined in the Reference Model to describe architectural features of their research infrastructures. The resulting document from EMSO is presented below.

How to Use the Reference Model

The European research infrastructure EMSO is a European network of fixed-point, deep-seafloor and water column observatories deployed in key sites of the European Continental margin and Arctic. It aims to provide the technological and scientific framework for the investigation of the environmental processes related to the interaction between the geosphere, biosphere, and hydrosphere and for a sustainable management by long-term monitoring also with real-time data transmission. Since 2006, EMSO has been on the ESFRI (European Strategy Forum on Research Infrastructures) roadmap; it entered its construction phase in 2012. Within this framework, EMSO is contributing to large infrastructure integration projects such as ENVRI and COOPEUS. The EMSO infrastructure is geographically distributed in key sites of European waters, spanning from the Arctic, through the Atlantic and Mediterranean Sea to the Black Sea. It is presently consisting of thirteen sites that have been identified by the scientific community according to their importance respect to Marine Ecosystems, Climate Changes and Marine GeoHazards.

The data infrastructure for EMSO is being designed as a distributed system. Presently, EMSO data collected during experiments at each EMSO site are locally stored and organized in catalogues or relational databases run by the responsible regional EMSO nodes. The EMSO data architecture is currently adapted to the ENVRI Reference Model. As shown in Figure 1, according to the ENVRI-RM it includes the 5 ENVRI common subsystems. Concepts and terms defined in ENVRI-RM are used to illustrate the currently practiced common data management strategies for real time as well as archived data within the EMSO distributed data management system.

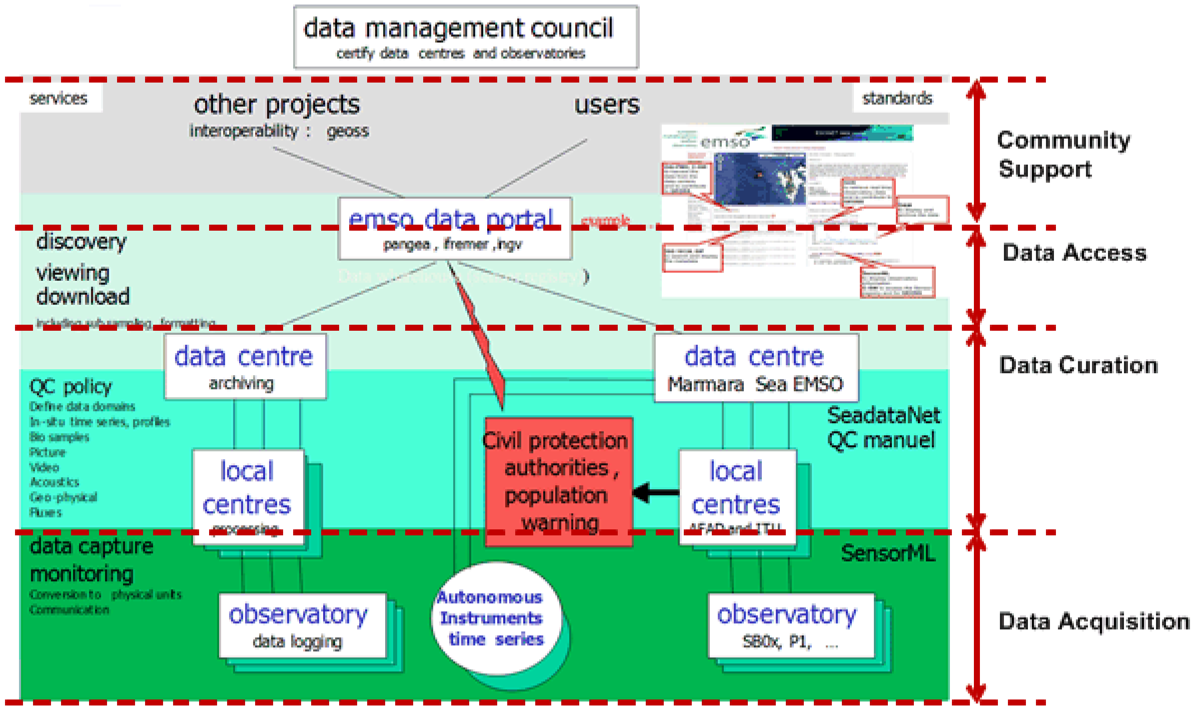


Figure 1: EMSO Distributed Data Management System

Data Acquisition

The EMSO **data acquisition sub-system** collects raw data from EMSO's marine observatories, which represent sensor arrays of varying geometry and various instruments or human observers, and brings the measures (data streams) into the system. **instrument configuration** and **design** is specified depending on the scientific demands and includes **sepcification of measurement**.

Depending on the deployment situation and nature of collected data, EMSO data is collected in real-time or delayed mode. Both **data collection** methods are performed by the regional nodes of EMSO that are responsible for the operation of marine observatories. Marine observatories have to deal with many technological challenges due to their extreme, deep sea deployment locations. Therefore data acquired by marine observatory sensor systems is most often temporarily staged within the instruments or the observatory’s internal storage systems, and real-time transmission of data is only provided by observatories that are connected by submarine cables or permanent satellite connections. Whereas real time data are immediately available, the staged data becomes available for these systems only after visits during dedicated ship expeditions when the instruments are recovered or maintained. In addition, data are acquired through laboratory studies performed on material or samples collected at marine observatory sites such as multidisciplinary analyses of water samples, sediment cores, tow or trap catches.

Depending on the instrumentation and observatory design, on-site quality control and data filtering is applied, generally followed by a transformation process which converts the instrument specific data format into a transmission format required by EMSOs ***data curation*** and ***processing*** systems at the regional data centre nodes. The data collected by the *data acquisition sub-system* are transmitted to the ***data curation*** , to be maintained and archived there.

Data Curation

The EMSO **data curation sub-system** facilitates **data curation**, **quality** and **preservation** of scientific data. It is operated at the data centres responsible for archiving the data acquired by the EMSO regional nodes. Three major data centres are currently offering these services for EMSO data: UniHB (PANGAEA), INGV (MOIST) and IFREMER (EUROSITES).

data importer are provided by these institutions which either transfer the above mentioned data transmission format into an archival format or provide editorial tools and interfaces to ingest delayed mode data and laboratory analysis into their systems. Data which are intended to be transferred to the regional nodes data archives are quality checked, linked with an appropriate set of **metadata** according to international standards and persistently identified, depending on the archives internal standards and procedures. EMSO offers **catalogue service** and **data exporter** for each regional node. The node systems PANGAEA and MOIST services based on metadata standards such as ISO19115, GCMD-DIF and extended Dublin Core, for EUROSITES data, metadata is extracted from NetCDF files via a central

EMSO service. **data exporter** are not yet fully implemented at all EMSO nodes, however it is planned to provide NetCDF export services for each node. The regional archives are responsible for cataloguing and long term preservation of these data that are provided for users via EMSO's *data access* and *discovery* subsystems.

Data Access and Discovery

The EMSO **data access sub-system** enables discovery and retrieval of data housed in data resources managed by a *data curation sub-system*. EMSO offers **discovery and access** via a common **metadata catalogue** and web portal which can be visited at http://dataportals.pangaea.de/emso. The portal is based on the brokerage system panFMP (http://www.panfmp.org) and uses Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) or simple file transfer via FTP/HTTP to harvest metadata from EMSOs distributed regional node data archives and their archival systems PANGAEA, MOIST and EUROSITES.

The EMSO data portal offers machine-human as well as machine-machine search facilities and discovery services based on the collected **metadata**. This includes a simple web-based user interface, a data search engine, which is offered at the EMSO data portal in a Google like style. In addition the data portal offers a common discovery service following the OpenSearch specification including the OpenSearch-Geo extension. A Open Geospatial Consortium (OGC) Catalogue Service for Web (CS-W) interface is currently under development.

A centralized data export service for these archived data is not implemented or planned, therefore, unless each EMSO data archive offers its own NetCDF data transformation service (see above) data requests are not yet processed by the EMSO data portal but are redirected to the hosting data archives which provide their own data access services for data retrieval.

Access to real time data is also offered via the EMSO data portal. EMSO has chosen to implement core standards of the OGC Sensor Web Enablement (SWE) suite of standards, such as Sensor Observation Service (SOS) and Observations and Measurements (O&M) to deliver real time data. These interfaces and formats are used to offer a common, web based SOS client which provides interactive visualizations of real time data.

Data Processing

Centralized **data processing sub-system** s that aggregate the data from various resources and provides computational capabilities and capacities for conducting data analysis and scientific experiments are not yet implemented for EMSO. Once more regional EMSO nodes and their data archives support NetCDF data export, it has been envisaged to introduce data visualization and plotting services at the EMSO data portal following the ESONET example. However presently, data processing services such as visualization, mining, as well as statistical services, are exclusively provided by each regional node and its responsible data centre.

Community Support

Centralized **community support sub-system** services to **profile managemen**t, control and **tracking** **users**' activities and supports users to conduct their roles in communities are not yet implemented or planned for EMSO.

Summary

The EMSO example demonstrates how to use the common language defined by the Reference Model in documentation to communicate with the RDA community.

It has been recognised there is a common challenge when communicating with external organisations or communities -- "*your 'model' is not my 'model', your 'data' is not my 'data'*". With a public accessible reference base, an external community who has little domain knowledge, such as the RDA, is able to understand the specific descriptions of EMSO by looking up the terminology in the Reference Model. In a way, using the Reference Model, the communication efficiency can be improved.

The ENVRI Reference Model provides a set of ready-to-use terminology, in principle:

- Terms in the Science Viewpoint can be used for describing requirements, use scenarios, and human activities;
- Terms in the Information Viewpoint for describing information objects handled in a system, their action types, constrains, states, and lifecycles; and
- Terms in the Computational Viewpoint for describing functionalities, computational components, interfaces and services.

A reader may have noticed there are some terms in the writing that are different from the ones linked back in the Reference Model. For example, " ***example3_setup*** *(… of each observatory)"* is linked to "***instrument configuration***". The intention is to show that in practice, to pursue the fitness, significance or beauty of the writing, an author may use different vocabularies to express the same concept. However, one can link them to the related concepts and definitions in the Reference Model to indicate the precise meanings. In this sense, using the Reference Model is different from using a dictionary – referring to the Reference Model places more emphasis on conceptual relativity.

Example 4: Using the Reference Model as design reference (EPOS)

Descriptions of the Example

Although it looks similar, this example provides a difference perspective on the usage of the Reference Model to that of Example 3: Using the Reference Model in documentation (EMSO).

The ENVRI Reference Model is characterised by being both an ontology and a model. While Example 3: Using the Reference Model in documentation (EMSO) demonstrates how to make use of its ontological framework in documentation, in this example, we exploit its representation as a model, which enables structural thinking and is more useful in the construction of an infrastructure and the organisation of design activities.

The European Plate Observing System (EPOS) is the European integrated solid earth sciences research infrastructure; a long-term plan to integrate existing national research infrastructures for seismology, volcanology, geodesy and other solid earth sciences. One of EPOS' goals is to provide the technical and legal framework by which to automate discovery and access to datasets and services provided by existing national (and trans-national) research institutions and monitoring networks throughout Europe. Another goal is to provide a standard set of core services by which researchers and other interested parties can interact with the federated infrastructure independently of the any particular data centre or national infrastructure. By providing such a common service interface and federation of resources, EPOS will be able to provide greater access to data recorded by existing and future monitoring networks, laboratory experiments and computational simulations, and foster greater cross-disciplinary research collaborations.

EPOS was included in the European Strategy Forum on Research Infrastructures (ESFRI) Roadmap in December 2008 and is currently in its Preparatory Phase; EPOS is scheduled to enter its Construction and Operational Phase in 2015.

EPOS is an infrastructure that intends to integrate several existing infrastructures which in the past have generally been constructed on a national scale only. There already exist established data centres with established working practices and monitoring networks. The challenge for EPOS is to provide a lightweight service layer that can be placed over these existing established infrastructures whilst disguising the underlying heterogeneity of components; this challenge is at least partially mitigated by the existence of certain protocols and data formats that are already standard in some parts of (for example) seismology, and a general drive within EPOS to further extend standardisation throughout its constituent institutions -- though it is not clear how extensively this level of standardisation will apply to all of the (currently highly disparate) earth sciences covered by EPOS' remit.

It is intended that ENVRI contribute in some way to the design of the EPOS Core Services, whether by the production of useful tools (via ENVRI WP4) or by the application of the ENVRI Reference Model (ENVRI-RM) for infrastructure layout and design (via ENVRI WP3). Focusing on the latter, ENVRI-RM should be able to simplify the design problem by breaking it down into well-defined subsystems of components specified from different complementary viewpoints (principally **Science Viewpoint**, **Information Viewpoint**and **Computational Viewpoint**).

How to Use the Reference Model

Following the guidence of the **ENVRI Common Subsystem** the EPOS design issues can be broken down as follows:

Acquisition

Data acquisition is performed by EPOS' constituent 'client' infrastructures; existing monitoring networks and laboratories, collected by data centres and presented for discovery and access to the EPOS integration layer. Many of these client systems operate in real-time (for example the continuous data streams produced by seismograph networks), requiring concurrently active data curation facilities (storage, persistent identification and metadata assignment).

Curation

Data is principally curated within existing data centres that publish their datasets according to some agreed protocol. These data centres have their own data collection policies, but EPOS intends to promote the adoption of common metadata in order to ease interoperation, based on a three-level model consisting of discovery metadata (using extended qualified Dublin Core) which is derived from contextual metadata (using CERIF, the Common European Research Information Format), which points to detailed metadata (domain-specific and associated with a particular service or resource). EPOS will also provide a global persistent identification mechanism for continuous data streams and discrete datasets (the latter possibly using the mechanisms produced by the EUDAT project).

Curation, Brokering and Processing

Given global persistent identification and metadata, as well as the use where possible of standard data formats, it is intended that tools be produced to search over and extract specific datasets from different sites based on geospatial (and other) requirements. This along with tools for modelling, processing, data mining and visualisation form the data-oriented integration layer of the EPOS Core Services. These sit atop the 'thematic layer' of the Services, which divide services by domain and forms (for example seismology, volcanology and geodesy as well as satellite data, hazard maps, geomagnetic observatories and rock physics laboratories).

Because EPOS is making a concerted effort to integrate data standards and services, the resultant infrastructure should be less reliant on the brokering model than otherwise expected; the homogenisation of resources means that it will not be so necessary to maintain interfaces between heterogeneous resources required to be interoperable.

EPOS also intends to provide access for researchers to high-performance computation facilities as provided by such infrastructure projects as PRACE.

Use

EPOS intends to provide training facilities to its research demographic; it is as yet unclear if EPOS intends to provide any kind of 'social' aspect to its core services (annotation of datasets, record of individual researchers' interactions with the infrastructure, etc.). It is a goal however of EPOS to promote best practices and reward participation, as well as to increase the visibility of research results produced using EPOS services. This implies that community support will become an increasingly important aspect of the EPOS infrastructure as the basic integration challenge it faces becomes solved.

Summary

Like EPOS, ESFRI Environmental Research Infrastructures are characterised as large-scale distributed complex systems involving numbers of organisations across different European countries. Design and implementations become large collaborative activities subject to change and are evolving, which bring significant challenges. Considering the difficulty of ensuring efficiency and productivity, it is not only what to do but how to do it that is important. We observe no approach is currently in use to assist the organisation of the design activities.

The ENVRI Reference Model captures common requirements of a collection of representative environmental research infrastructures, providing a projection of Europe-wide requirements they have, which in potential can be served as a technology roadmap to position and orchestrate collaborations in design and developments. It provides well-defined subsystems of components specified from different complementary viewpoints (Science, Information and Computation), which can help break down the complexity and simplify the design problems, enabling designers to deliver a practical architecture that leads to concrete implementations. It offers a descriptive framework for specifying uniform distributed systems, allowing designers from different organisations to carry out design activities in parallel.

EPOS/ENVRI modelling

EPOS/ENVRI Modelling.

EPOS/ENVRI Modelling

A (very) generic overview of EPOS data access and (RI-internal) processing in terms of ENVRI-RM computational objects.

Two current possible avenues of development:

- **Instantiation** – identifying specific services created or in the process of being created that are instances of the computational objects above.
- **Case study**– take a usecase (one out of FutureVolc?) that allows a story to be told about how the user could interact with data via EPOS in terms of the ENVRI-RM.

Example 5: Using the Reference Model to explain the technology details of common services (WP4 practices)

Descriptions of the Example

ENVRI working package 4 responses to deliver common services to support the constructions of ESFRI ENV RIs. Initially, the implementations focus on a **data access subsystem** that supports integrated data discovery and access. In order to help ESFRI project managers, architects, and developers understand the design and implementation of these services, this example uses the terms and concepts from the Reference Model to explain the technology details of these services.

How to Use the Reference Model

We start with the semantic harmonisation service developed by the team in Task 4.2 [**39**]. The development is conducted to support the use case "Iceland Volcano Ash". The goal is to support scientists to analyse Iceland behaviour using data provided by different research infrastructures during a specific time period.

Science Viewpoint

Defined by the Reference Model Science Viewpoint, the semantic harmonization is a behaviour belong to the data publication community, which captures the business requirements of unifying similar data (knowledge) models based on the consensus of collaborative domain experts to achieve better data (knowledge) reuse and semantic interoperability.

Computational Viewpoint

A data publication community interacts with a data access subsystem to conduct user roles. The computational specification of the data access subsystem is given in Figure 1. The model specifies a **data access subsystem** which provides data broker that act as intermediaries for access to data held within the data curation subsystem, as well as semantic brokers for performing semantic interpretation. These brokers are responsible for verifying the agents making access requests and for validating those requests prior to sending them on to the relevant data curation service. These brokers can be interacted with directly via virtual laboratories such as experiment laboratories (for general interaction with data and processing services) and semantic laboratories (by which the community can update semantic models associated with the research infrastructure).

Figure 1: Computational specification of data access subsystem

Definitions

A **data broker** object intercedes between the data access subsystem and the data curation subsystem, collecting the computational functions required to negotiate data transfer and query requests directed at data curation services on behalf of some user. It is the responsibility of the data broker to validate all requests and to verify the identity and access privileges of agents making requests. It is not permitted for an outside agency or service to access the data stores within a research infrastructure by any means other than via a data broker.

An **experiment laboratory** is created by a science gateway in order to allow researchers to interact with data held by a research infrastructure in order to achieve some scientific output.

A **semantic broker** intercedes where queries within one semantic domain need to be translated into another to be able to interact with curated data. It also collects the functionality required to update the semantic models used by an infrastructure to describe data held within.

A **semantic laboratory** is created by a science gateway in order to allow researchers to provide input on the interpretation of data gathered by a research infrastructure.

Please click the links to find out the specification details of these computational objects and the interactions between them.

The implementation conducted by WP4 T4.2 is an ***instantiation*** of the above computational objects specified in the Reference Model, that uses existing software components and developed approaches to enable integration and harmonization of data resources from cluster's infrastructures and publication according unifying views.

Figure 2 depicts the computational components deployed in the prototype implementation. The service receives users' requests via the SPARQL-endpoint. Then, it can automatically retrieve and integrate real measurement data collections from distributed data sources. The current prototype focuses on datasets from two different ESFRI projects:

- ICOS, which is organized by atmospheric stations which perform measurements of the CO2 concentration in the air and
- EURO-Argo observations that were provided in separate collections grouped according to the float that performed measurements of the ocean temperature.

The prototyped service uses two semantic models to provide mapping between representations: the RDF Data Cube vocabulary and the ENVRI vocabulary. The ENVRI vocabulary is derived from the OGC and ISO "Observations & Measurements" standard (O&M) , SWEET and GeoSparql Vocabulary.

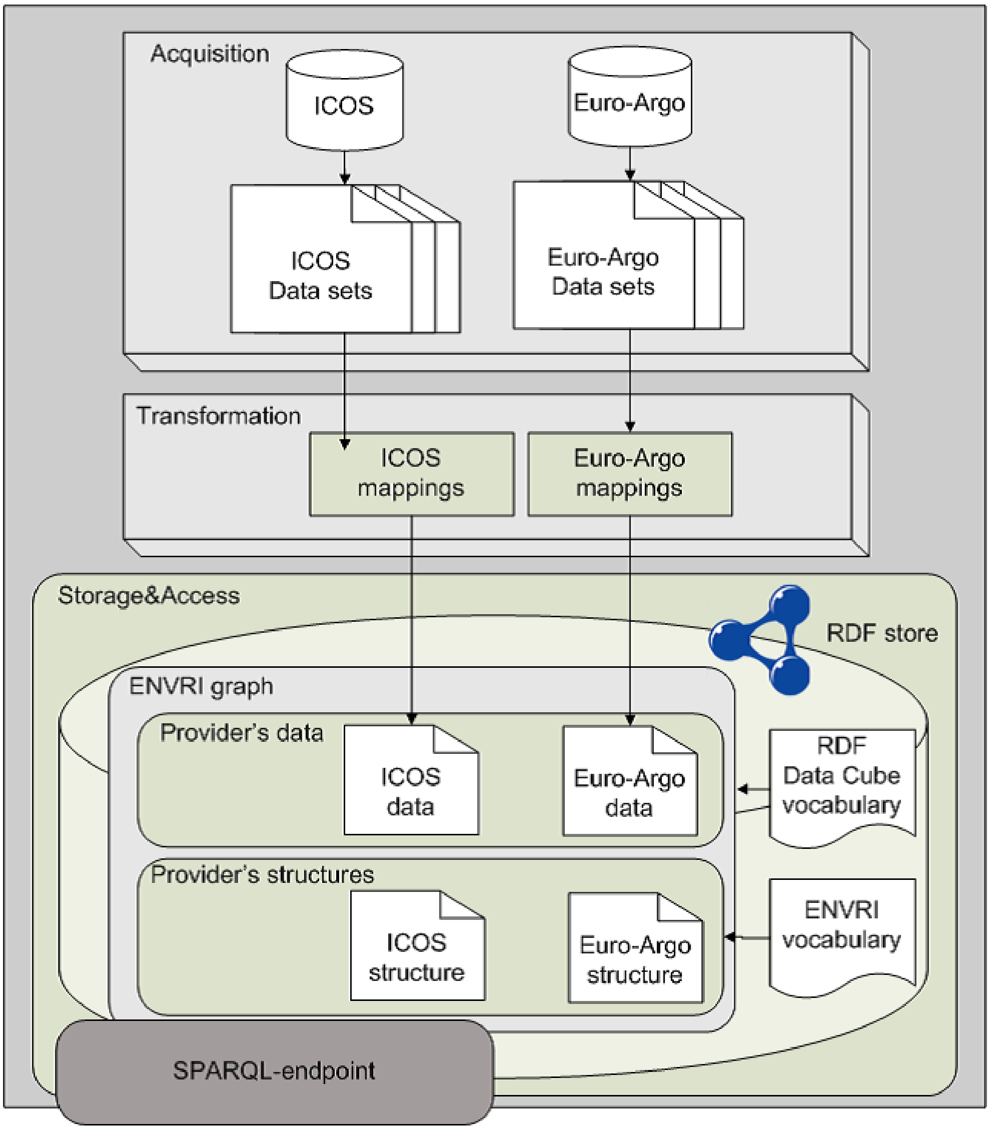


Figure 2: The Deployed service components for semantic harmonization [39]

Table 1 provides the mapping between Reference Model computational objects and the deployed service components. Among them, the *Transformation* component serves as a **data broker** to negotiate data access with data stores within heterogeneous research infrastructures. An (instance of the) **semantic broker** is implemented using the RDF store technology which provides the semantic mappings and translations.

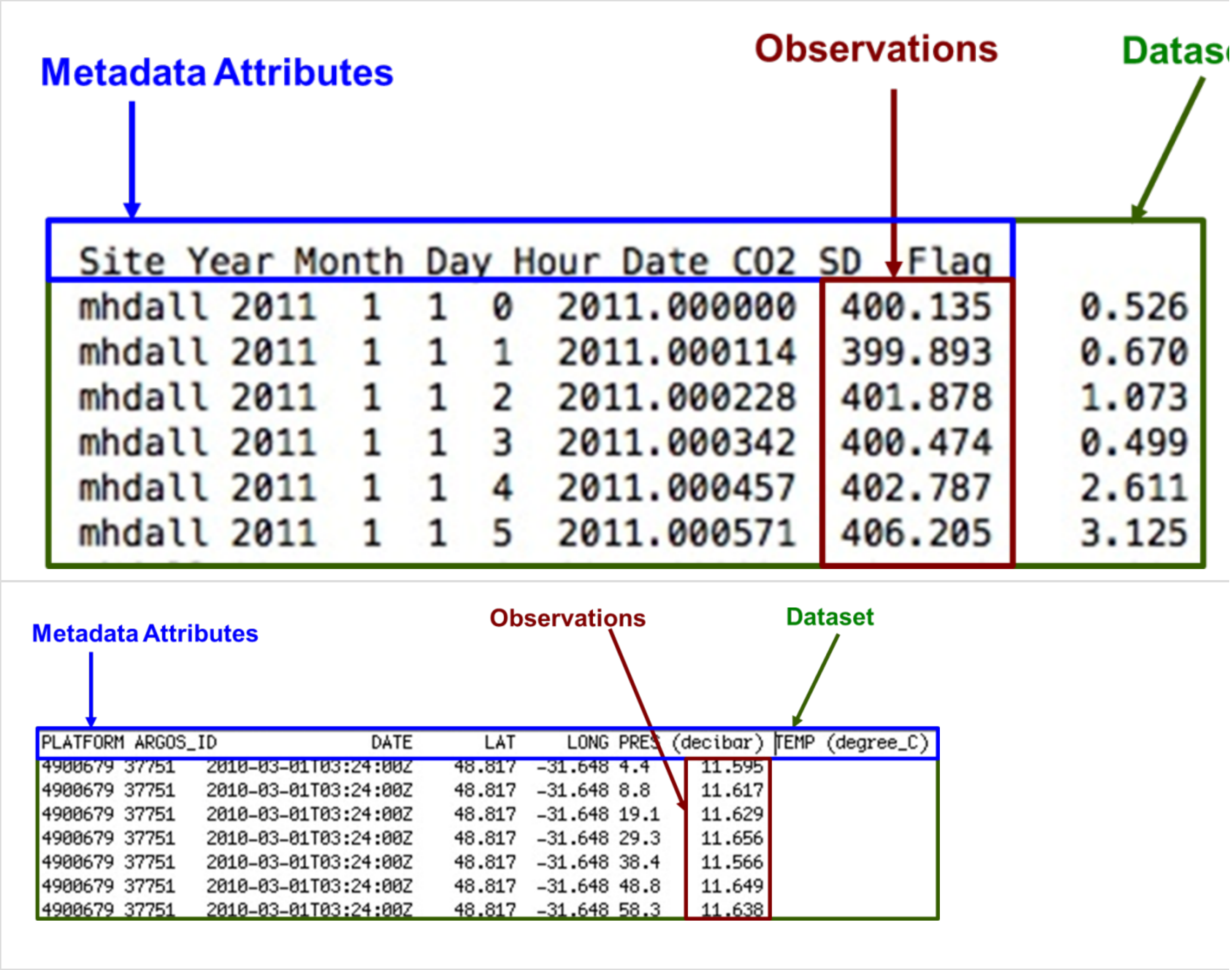
Table 1: Mapping of the deployed service components to the Reference Model computational objects

RM Computational Objects	Deployed Service Components
Data Broker	Transformation (ICOS mappings, EuroArgo Mappings)
Experiment Laboratory	SPARQL-endpoint
Semantic Broker	Provider's data (ICOS data, EuroArgo data) Provider's structures (ICOS structure, EurArgo structure)
Semantic Laboratory	RDF Data Cube Vocabulary, ENVRI Vocabulary

In the following, we explain the design of the information model of the semantic harmonisation service.

Information Viewpoint

Analysing the environmental data schema results in identifying the common structural concepts, the ENVRI vocabulary, which include the terms such as "metadata attributes", "observation", "dataset". Data retrieved from the different sources are firstly mapped to this uniform semantic model. Figure 3 gives two examples, and shows how datasets of ICOS and EuroArgo can be mapped to the ENVRI vocabulary, respectively.



Site	Year	Month	Day	Hour	Date	CO2	SD	Flag
mhdall	2011	1	1	0	2011.000000	400.135	0.526	
mhdall	2011	1	1	1	2011.000114	399.893	0.670	
mhdall	2011	1	1	2	2011.000228	401.878	1.073	
mhdall	2011	1	1	3	2011.000342	400.474	0.499	
mhdall	2011	1	1	4	2011.000457	402.787	2.611	
mhdall	2011	1	1	5	2011.000571	406.205	3.125	

PLATFORM	ARGOS_ID	DATE	LAT	LONG	PRES (decibar)	TEMP (degree_C)
4900679	37751	2010-03-01T03:24:00Z	48.817	-31.648	4.4	11.595
4900679	37751	2010-03-01T03:24:00Z	48.817	-31.648	8.8	11.617
4900679	37751	2010-03-01T03:24:00Z	48.817	-31.648	19.1	11.629
4900679	37751	2010-03-01T03:24:00Z	48.817	-31.648	29.3	11.656
4900679	37751	2010-03-01T03:24:00Z	48.817	-31.648	38.4	11.566
4900679	37751	2010-03-01T03:24:00Z	48.817	-31.648	48.8	11.649
4900679	37751	2010-03-01T03:24:00Z	48.817	-31.648	58.3	11.638

Figure 3: Datasets as provided by ICOS (above) with CO2 concentrations and by EURO-Argo (below) with ocean temperature measurements

Semantic mappings are based on observation statements. For example, the following observation statement declares the measurements about "**air**":

"*Observation of the CO2 concentration in samples of air at the Mace Head atmospheric station which is located at (53_20'N, 9_54'W): CO2 concentration of the air 25m above the sea level on Jan 1st, 2010 at 00:00 was 391.318 parts per million*".

"**Air**" is represented as the concept of air in GEneral Multi-lingual Environmental Thesaurus (GEMET) by assigning the URI to it (entity naming). The GEMET concept of air is then defined as an instance of envri:FeatureOfInterest (entity typing).
The mapping rules are specified by using the Data cube plug-in for Google Refine. The mappings are executed to obtain RDF representations of the source data files. As such they are uploaded to the Virtuoso OSE RDF store and are ready to be queried at a SPARQL-endpoint.
The data harmonization process described above is captured by the Reference Model. As shown in Figure 4, the Information Viewpoint models the mapping of data according to **mapping rules** which are defined by the use of **local** and **global conceptual model**. Ontologies and thesauri are defined as **conceptual models**, and those widely accepted models such as, GEMET, O&M, Data Cube, are declared global conceptual models whereas the ENVRI vocabulary is specified as a local one, because it has been developed within the current project without being yet accepted by a broad community.

Figure 4: The RM Information specification related to the semantic harmonisation

Describing a process using the ENVRI Reference Model concepts is to instantiate the concepts that can be mapped to the process. Figure 5 illustrates the instantiation (all boxes with a dashed line) of the ENVRI Reference Model concepts focusing at the harmonization process described above. The same could be demonstrated for the EuroArgo dataset with the feature of interest being ocean. For each part of the observation mapping rules have to be defined to be able to query both datasets at a certain time period.

Figure 5: Mapping of the deployed information model with that of the the Reference Model

The tables below show the mapping between the harmonisation process and the concepts in the ENVRI RM information viewpoint. The example shows that both bottom up (from the applied operation to the model description) and top down approaches (from the model definitions back to the applied solution) can lead to a better understanding of the Reference Model itself and of how components should work properly in a complex infrastructure.

Table 2: Mapping between the Reference Model **Information objects** and those in the deployed service

Information Object in RM	**Component/Object in Task 4.2**
specification of measurements or observations	Observation of the CO2 concentration in samples of **air** at the Mace Head atmospheric station which is located at (53_20'N, 9_54'W): CO2 concentration of the air 25m above the sea level on Jan 1st, 2010 at 00:00 was 391.318 parts per million
mapped	GEMET:245 is instance of FeatureOfInterest class
conceptual model	GEMET, O&M, DataCube
conceptual model	ENVRI vocabulary
local concept	FeatureOfInterest (ENVRI vocabulary)
global concept	Component Property, GEMET:245, FeatureOfInterest (O&M)
mapping rule	GEMET:245 create as instance of FeatureOfInterest class
published	ICOS data CO2 of air, EuroArgo data ocean temperature

Table 3: Mapping between the Reference Model **Action Types** and those in the deployed service

Information Action Tyoes in RM	**Operation in Task 4.2**
build conceptual models	Build ENVRI vocabulary as extension of DataCube and on basis of O&M concepts
setup mapping rules	Define rule: GEMET:245 create as instance of FeatureOfInterest class
perform mapping	Perform Mapping using Google Refine
query data	SPARQL query: http://staff.science.uva.nl/~ttaraso1/html/queries/Q1.rq

Summary

This example demonstrate the feasibility of the design specifications of the reference model. Instances of selected model components can be developed into common services, in this case, a **subsystem** that supports integrated data discovery and access. Data products from different environmental research infrastructures including, measurements of deep sea, upper space, volcano and seismology, open sea, atmosphere, and biodiversity, can now be pulled out through a single data access interface. Scientists are using this newly-available data resource to study environmental problems previously unachievable including, the study of the climate impact caused by the eruptions of the Eyjafjallajökull volcano in 2010.

Example 6: Using the Reference Model to provide the external advice to the ICOS RI Design Studies

The Integrated Carbon Observatory System, ICOS, is built to enable research to understand the greenhouse gas budgets and perturbations. New Carbon Portal is being designed and envisioned as a virtual data centre, to provide a single access point for environmental scientists to discover, obtain, visualise and track observation measures produced from the observation stations as quick as possible.

The design of the ICOS Carbon Portal is challenged by the complicated requirements of dataflow from the acquisition of the measures to the processing and publication of the data products. The ICOS national observation stations are highly distributed; data are semantically diverse, processes are different from nation to nation, organisation to organisation; measurements are varied from experiments to experiments.

Since January 2014, the ENVRI Reference Model team has been supported ICOS to examine the requirements and optimise the design using the Reference Model concepts and framework. The Reference Model has been firstly introduced to the ICOS architects. On 27th-28th J an 2014, a ICOS-Reference Model workshop was held in Cardiff, providing the training of the Reference Model, and assisting the ICOS architects to analyse the design of the Carbon Portal. The Reference Model contributes to the ICOS system design by simplifying the design problem, breaking it down by subsystem, providing a uniform framework with well-defined subsystems of components specified from different complementary viewpoints (Science, Information and Computation), which promotes structural thinking in the construction of system architectures. This use of the Reference Model enables designers to deliver a practical architecture that leads to concrete implementations. The initial benefit/cost analysis shows that using the Reference Model, the design cost of the ICOS Carbon Portal could be reduced, and future additions to the ICOS can be more easily implemented.

The positive feedback from the ICOS architects led to the communications between the ICOS head office. On 13 March, in a meeting held in Helsinki, the Reference Model was able be presented to the ICOS director and the head office, who finally decided and pushed the adoption of the Reference Model within the community. Shortly after, the Reference Model team organised training events to the ICOS community, and delivered comprehensive analysis and design specification for the ICOS system, including the explicit definitions of the ICOS community, responsibilities of each role, specification of data lifecycles and actions of the data step by step, computation model, service interfaces and interactions.

On 4 Jun 2014, a workshop is held in London Heathrow welcomed all key organisations of ICOS Research Infrastructure representing the ICOS Head Office, Thematic Centres (TC), Central Analytical Labs (CAL), and Carbon Portal(CP). The analysis results of the ICOS system using the Reference Model was presented to the community, and received encouraging feedback. As the director of ICOS, Werner, said, the Reference Model helped ICOS clarify own thinking, identified many important issues, and increased internal cooperation.

As a follow-up action, a second workshop was organised on 10th September, Amsterdam, where Reference Model team was invited for further discussion and collaborations. The workshop reviewed the operation workflows in different thematic centres and the Central Analytical Lab. Using the Reference Model concepts and principles, the ICOS stakeholders, ETC, ATC, and CAL, are able to provide detailed workflows which explicitly describe the process steps from data collection, quality checking, data archiving, to the publication of the ICOS data products. The discussions also identified important design issues need to be resolved, e.g. what PID mechanism to be used; how many PIDs ICOS needs; what (data or metadata) needs to be pointed; what is L0, L1 and L2 data, how states changes as the results of operational actions, etc.

This report provide requirement analysis and design advice for ICOS research infrastructure using ENVRI Reference Model as analysing tool.

Other references includes a progress report to ICOS Interim Scientific Advisory Board (by Werner), and Analysis of ICOS RI Using Reference Model_v1_17Sep_Updates of SV.doc after the Amsterdam discussions.

Analysis of ICOS Research Infrastructure from *Science Viewpoint*

ICOS Research Infrastructure (ICOS RI) is built to provide the long-term observations required to understand the present state and predict future behaviours of climate, the global carbon cycle and greenhouse gases emissions.

ICOS RI Objectivise: include

- Tracks carbon fluxes in Europe and adjacent regions by monitoring the ecosystems, the atmosphere and the oceans through integrated networks.
- Provides the long-term observations required to understand the present state and predict future behaviour of the global carbon cycle and greenhouse gas emissions.
- Monitors and assesses the effectiveness of carbon sequestration and/or greenhouse gases emission reduction activities on global atmospheric composition levels, including attribution of sources and sinks by region and sector.

Figure 1 shows the annotations of ICOS RI organisational structure using the Reference Model (RM) terminologies from the Science Viewpoint. From the analysis, the community roles and behaviours can be identified, and workflow can be understand.

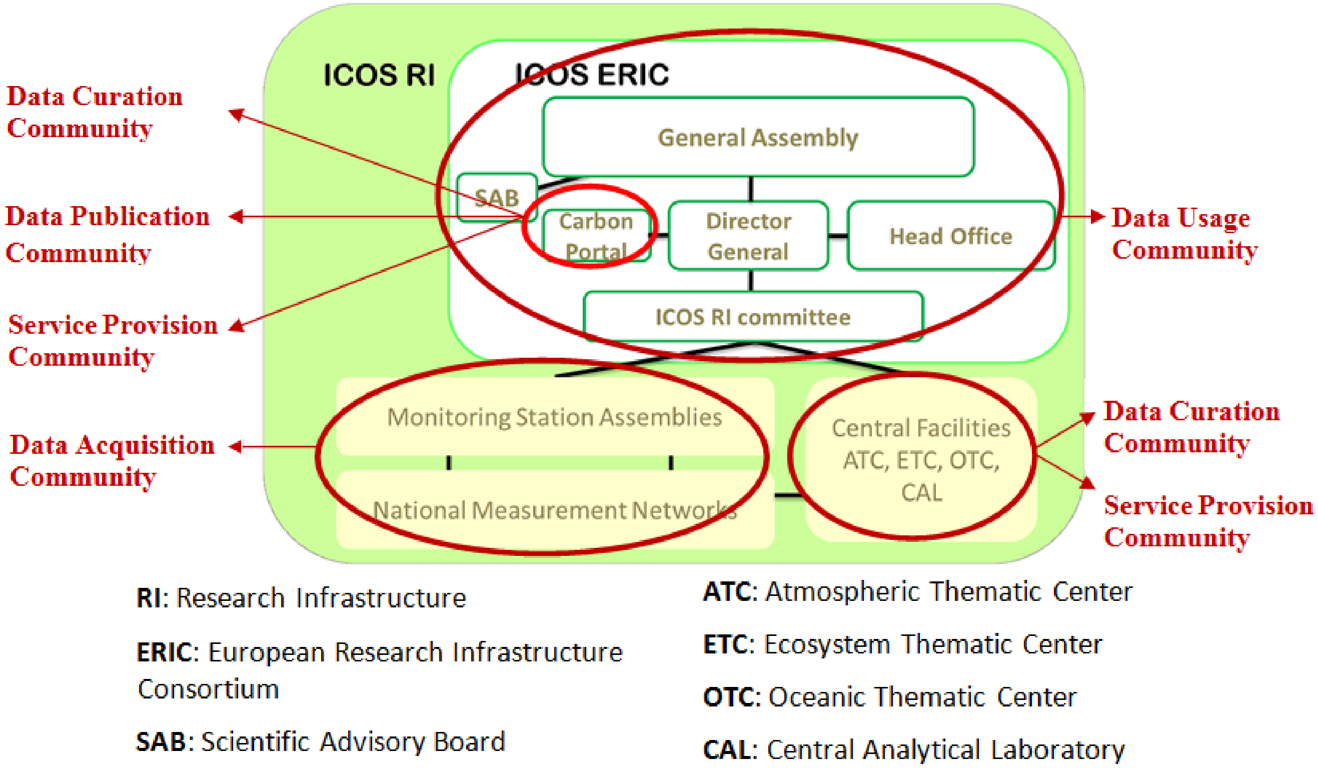

ICOS RI
ICOS ERIC
General Assembly
SAB
Carbon Portal
Director General
Head Office
ICOS RI committee
Monitoring Station Assemblies
National Measurement Networks
Central Facilities ATC, ETC, OTC, CAL
Data Curation Community
Data Publication Community
Service Provision Community
Data Acquisition Community
Data Usage Community
Data Curation Community
Service Provision Community
RI: Research Infrastructure
ERIC: European Research Infrastructure Consortium
SAB: Scientific Advisory Board
ATC: Atmospheric Thematic Center
ETC: Ecosystem Thematic Center
OTC: Oceanic Thematic Center
CAL: Central Analytical Laboratory

Figure 1: Annotation of ICOS Organisational Structure (1) Using Terminology of the Reference Model *Science Viewpoint*

ICOS RI Roles

Table 1 provides the roles identified in ICOS Research Infrastructure, the descriptions of them, and the *role names* defined by the Reference Model.

Table 1: Roles in ICOS RI and Role Names in the Reference Model

Roles Instances in ICOS RI	Descriptions	RM SV_Roles Names
ICOS General Assembly		- Police or Decision Maker
Scientific Advisory Board (SAB)		- Police or Decision Maker
ICOS RI Committee	It is an advisory body to the Director of ICOS ERIC, and decides about strategies concerning the Carbon Portal.	- Police or Decision Maker
Director General		- Police or Decision Maker
Head Office/Headquarter (HO)	The ICOS RI Head Office will have three main task groups, which ar: 1. Managing the ICOS ERIC legal entity 2. Strategic scientific and technical planning, coordination and integration. 3. Community building, outreach, promotion and training	- Police or Decision Maker
Carbon Portal (CP)	The Carbon Portal shall provide a "one-stop shop" for ICOS data products. It is envisioned as a place where all data produced within ICOS station network can be discovered and accessedand where the scientific community can post elaborated data products that are obtained from ICOS data.	- Data Curation Subsystem - Data Access Subsystem - Service Provider Potentially, CP may also be: - Service Registry - PID Generator - PID Registry - Semantic Mediator
Connect projects and International network	Provide data to ICOS RI	- Data Originator

- Global networks GEOSS - Greenhouse gas flux assessment International programs	Consume the data provided by ICOS RI	- Data Consumer
The Central Analytical Laboratory (CAL)	CAL ensures the accuracy of observational data, thorough quality control and routine testing of air sampling material. It provides reference gases for calibration of in-situ measurements performed at the continuous monitoring stations. It also analyses air samples collected at the monitori ng stations. CAL is hosted by Germany.	- Data Curator
The Atmospheric Thematic Centre (ATC)	ATC is responsible for continuous and discontinuous air sampling, instrument development/servicing, data processing and storage. A central place is needed to ensure that all data are treated with the same algorithms and properly archived for the long term, that the ICOS atmo spheric stations can receive permanent support for optimal operation during their lifetime, and that new sensors can be smoothly implemented in the network in the future. ATC is coordinated and hosted by France, with Nordic Hub and Mobile Lab hosted by Finland.	- Data Curator - Data Curation Subsystem - Storage Administrator - Storage - Data Originator

The Ecosystem Thematic Centre (ETC)	ETC coordinates the ICOS Ecosystem Network providing assistance with instruments and methods, testing and developing new measurement techniques and associated processing algorithms. It also ensures a high level of data standardization, uncertainty analysis and database services in coordination with the ICOS Carbon Portal. ETC is coordinated and hosted by Italy, together with Belgium and France.	
The Ocean Thematic Centre (OTC)	OTC will be coordinating measuring the carbon cycle in oceans within ICOS. It will provide support to the ICOS marine network in the form of information and technical backup on the state of the art instrumentation and analytical methods. It will provide of data storage and processing techniques, quality control, and network-wide integration of data to into useful products, such as maps of CO_2 fluxes, carbon transport, and the assessment of ocean acidification.	

Monitoring Station Assemblies (MSA)	MSAs discuss technical and scientific matters, and services concerning their component to further develop and improve ICOS and its networks. MSAs work together with ATC, ETC and OTC, but have also independent role. MSA Members are scientific and technical experts from the monitoring stations of Member countries that constitute the basis of ICOS ERIC; All Atmospheric station PIs, Ecosystem station PIs and Ocean station PIs are the members of the respective MSAs.	• Environmental Scientist • (Measurement Model) Designer
Station Principal Investigators (SPI)		• Data Curator
Atmospheric Stations	They are established to measure continuously the greenhouse gas (CO2, CH4, N2O) concentration variability due to regional and global fluxes.	• Sensor • Sensor network • Technician • Measurer • Data collector • Data Acquisition Subsystem
Ecosystem Stations	They are built for monitoring the functioning of land ecosystems and the exchange of energy and greenhouse gases between the ecosystems and the atmosphere.	
Ocean Ships and Stations	Marine ICOS will provide the long-term oceanic observations required to understand the present state and predict future behaviour of the global carbon cycle and climate-relevant gas emissions.	
Users of ICOS data products: • Researchers; • International and national Operational Centres assimilating atmospheric composition data; • Policymakers and stakeholders involved in negotiating carbon reduction policies; • Carbon trading communities; • Regional authorities and carbon inventory agencies; • Private land owners and industrial contributors of greenhouse gas emissions; • The general public interested in greenhouse gas emissions and global climate change. • Commercial users • Others		• Scientist or Researcher • Police or Decision Maker • Private Sector (Industry investor or consultant) • General Public, Media or Citizen (Scientist)

ICOS RI Communities Behaviours

Table 2 provides mapping of ICOS roles to the ENVRI *5-common-community*. Analysing the role key responsibilities results in the mapping of the *community behaviours* defined in the Reference Model.

Table 2: Mapping of ICOS RI roles into the ENVRI Common Communities and Identifying the Community Behaviours

	Roles Instances in ICOS RI	Key Responsibilities	RM SV_Community Behaviours

Data Acquisition Community	• National Measurement Networks • Atmospheric Stations • Ecosystem Stations • Oceanic Ships and Stations	• Perform measurements according top ICOS standards • Collect data and send to Thematic Centres • Can have non-ICOS functionality & responsibilities, e.g., they may also • Collect other types of data • Perform their own data analysis (*not* official ICOS!) • Operate their own web sites	• Instrument Configuration • Instrument Calibration • Data Collection
	• Monitoring Station Assemblies (MSAs)	(See role descriptions)	• Design of Measurement Model
Data Curation Community	• Station Principal Investigators (SPIs)	• Perform quality checks • In near real time (for some systems) • After (pre-) processing at Thematic Centres • Before “final” datasets are “published”	• Data Quality Checking
	Central Facilities • Ecosystem Thematic Centre • Atmospheric Thematic Centre • Ocean Thematic Centre	• Compose and maintain procedures and protocols for measurements • Create “publishable” data sets • Keep own competence up to date • Maintain their own websites • Info on measurements • Near Real-Time data visualization • Data processing info (for SPIs, mainly) • Serve as experts • For stations within ICOS RI network • For external partners (if resources allow)	• Data Preservation • Data Product Generation • Data Replication
	• The Central Analytical Laboratory	(See role descriptions)	• (Instrument) Calibration • Data Quality Checking
	• Connect projects and International network	(See role descriptions)	
	• ICOS Carbon Portal	Organize and ensure back-up storage and long-term archiving of published ICOS data sets	• Data Replication • Data Preservation
Data Publication Community	• ICOS Carbon Portal	• Generate and provide effective tools to **publish**, **discover**, **access** and **retrieve** ICOS observations data according to user needs • Offer user-friendly, web-based **access** to products elaborated from ICOS data • Establish **interfaces** with other relevant data portals • Ensure basic **semantic interoperability** by maintaining a full copy of the standard metadata and data description documents (ontologies) held at the ICOS TCs, including the compilation of the vocabularies in use within ICOS • Coordinate **regular publication** of the ensemble of the ICOS data, with the TCs and the ICOS community of PIs • Organize **the traceability of downloaded** ICOS data, including the application of persistent unique identifiers for **citation** purposes • **Record** relevant bibliometric information and establish indicators about the use of ICOS data	• Data Publication • Data Discovery & Access • Semantic Harmonisation • Data Citation
Data Service Provision Community	• Central Facilities • Ecosystem Thematic Centre • Atmospheric Thematic Centre • Ocean Thematic Centre • Analytical Laboratory	• Process data (and analyze some samples)	

	• ICOS Carbon Portal	• Define and implement advanced web services and procedures for web-based data **visualization**, retrieval and **processing** • Encourage, **coordinate**, **facilitate** and ensure the operational provision of elaborated products and synthesis efforts based on ICOS data	• Service Description • Service Coordination • Service Composition
Data Usage Community	• ERIC Head Office	Organise general ICOS outreach actions on the basis of the scientific material (advanced data plots and visuals) provided by the Carbon Portal.	
	• Director General • ICOS RI Committee • ICOS Council • Scientific Advisory Board (SAB) • General Assembly	(See role descriptions)	
	• Global networks GEOSS • Greenhouse gas flux assessment International programs	(See role descriptions)	
	• Users of ICOS data products	(See role descriptions)	
	• ICOS Carbon Portal	Implement a common user **registration** & **authentication** system for ICOS that allows **usage tracking**	• User Profile Management • User Behaviour Tracking

Note

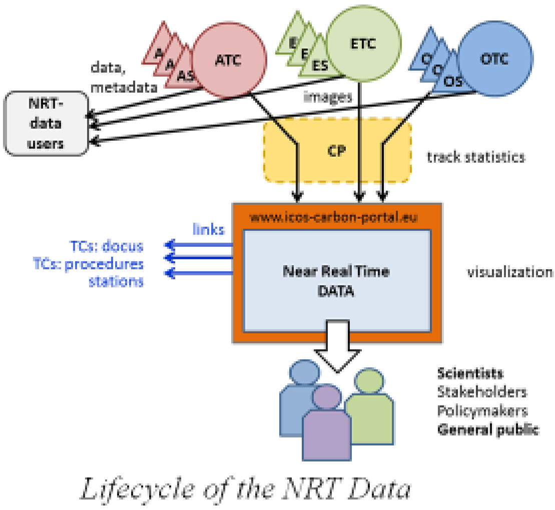
By ODP/RM definition, a computational system could play a passive role in a community. For example, ICOS Carbon Portal is regarded as a role in the communities of: Data Curation, Data Publication, Service Provision and Usage.

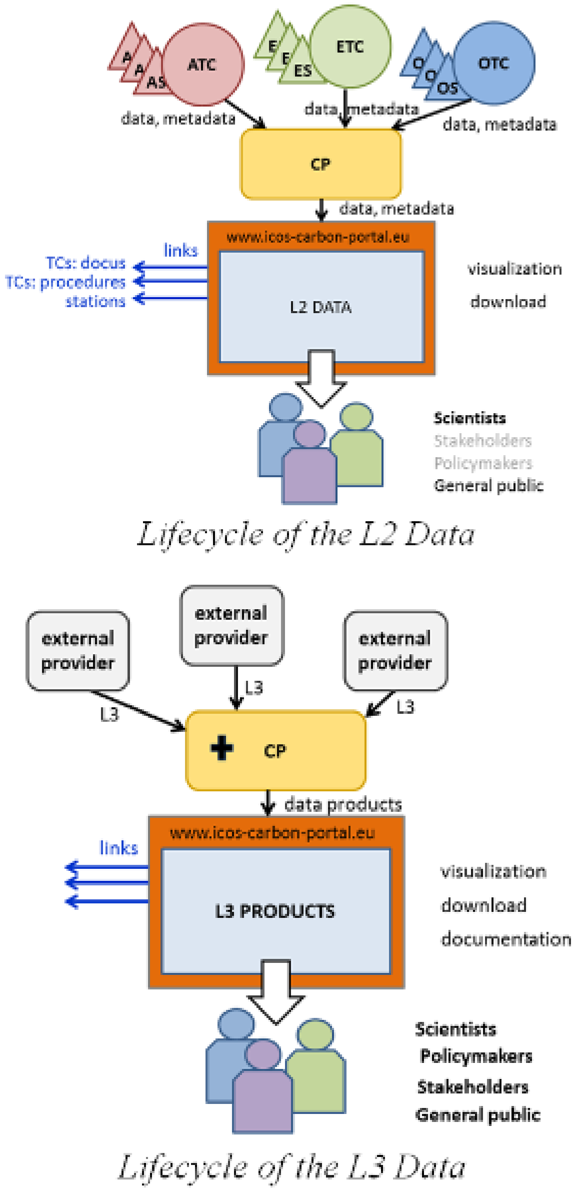
ICOS RI Workflow

We have explicitly defined the ICOS community roles and clearly described their behaviours. In this subsection, we examine how those roles interact with each other (through ICOS RI) and collaboratively fulfil the community objectives. We will output the result of the analysis in a workflow diagram which depicts the activities and processes conducted by roles, and the directions of controls and objects flows from one role to another.

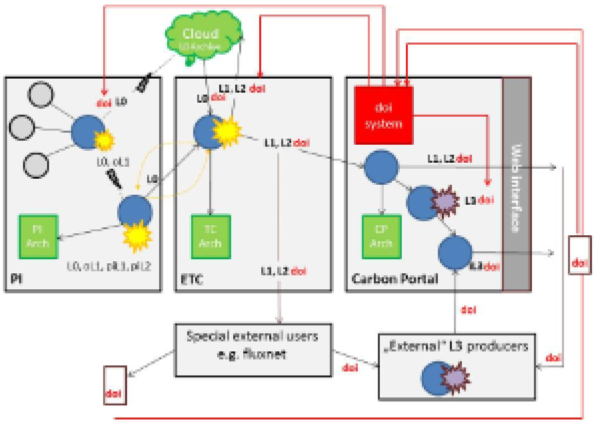
Figure 2 are the information obtained from ICOS team. (a) gives an overview of computation and data-flow in the ICOS RI. (b) provides the details of ICOS data life-cycle, and (c) describes the DOIs assigning process. From these information, we conclude the key community processes and workflow in Figure 3.

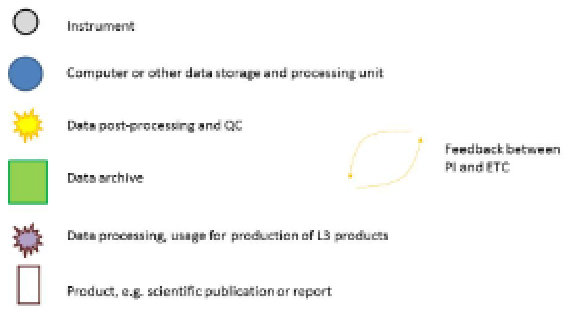
(a) An overview of the proposed data-flow in ICOS (April 2014)





(b) ICOS Data Lifecycle




Instrument
Computer or other data storage and processing unit
Data post-processing and QC
Data archive
Feedback between PI and ETC
Data processing, usage for production of L3 products
Product, e.g. scientific publication or report

(c) ICOS PIDs (DOIs) Assigning Process. DOIs by the Reference Model definition is one type of Persistent Identifiers (PIDs). (c) shows a DOI system will be established within the Carbon Portal to assign DOIs to the L0 data generated at the Stations, L1, L2 data produced at the Thematic Centres, and L3 data processed at the Carbon Portal.

Figure 2: Analysis of ICOS Requirements

Figure 3: ICOS RI Community Process. Each column corresponds to one ICOS RI community role. A black dot represents the starting point of the workflow, and a black dot with a circle represents an ending point of the workflow. Each box in a role column represents a process performed by that role. An arrow indicates the direction of the (control/object) flow between processes.

Figure 3 describes the workflow and the key community process from data collection to data access. The workflow starts from the process that each Station 1) “*collects the L0 data*”, and 2) “*stores the L0 data*”. At this point, each station may request the Carbon Portal to 3) “*generate PIDs (DOIs) for L0 data*”. With available PIDs, each station will 4) “*add PIDs (DOIs) to L0 data*”, also 5) “*add and store metadata for L0 data*”. Then, station Principle Investigators (PIs) will 6) “*check quality of L0 data*”. Thereafter, L0 data will be delivered to Thematic Centres. Each Thematic Centre will 7) “*store L0 data and metadata*”, and 8) “*archive L0 data and L0 metadata*”. Each Thematic Centre also 9) “*enables the visualisation of (the L0) data*”, to allow 10) end users to “*view (the L0) data from Thematic Centres websites*”. After 7), Thematic Centres also 11) “*pre-process L0 data to generate L1, L2 data*”, 12) “*store L1, L2 data*”, 13) request the Carbon Portal to “*generate PIDs (DOIs) for L1, L2 data*”, 14) ”*add PIDs (DOIs) to L1, L2 data*”, and 15) “*add and store metadata for L1, L2 data*”. At this point, station PIs may need to 16) “*check quality of L1, L2 data*”. After that, Thematic Centres will 9) “enable the visualisation of (the L1) data ” and 10) allow end users to “*view (the L1) data from Thematic Centres websites*”. Meantime, a copy of dataset will be sent to the Carbon Portal. The Carbon Portal will 17)“*archive L1, L2 data and L1, L2 metadata*”, 18) “*store L2 data*”, 19) “*enable search & discovery of L2 metadata*”, 20) “*enable download and visualisation of L2 data*”. This will enable end user to 21) “*view the L2 data from* www.icos-carbon-portal.eu”. The Carbon Portal will also 22) “track statistics” of any usage of the data. With stored L2 data, after 18), the Carbon Portal also 23) “*processes L2 data to generate L3 Data*”, 24) “*stores L3 data*”, 25) “*generates PIDs (DOIs) for L3 data*”, 26) “*adds PIDs (DOIs) to L3 data*”, 27) “*adds and stores metadata to L3 data*”, and 28) “*archive L3 data and L3 metadata*”. Meantime, the Carbon Portal also 19) “*enables search & discovery of (the L3) metadata*”, 20) “*enables download and visualisation of (the L3) data*”, which will enable end user to 21) “*view (the L3) data from* www.icos-carbon-portal.eu”. Again, any usage of the ICOS data will be 22) “*tracked*” by the Carbon Portal.

Analysis of ICOS Research Infrastructure from *Information Viewpoint*

The Information Viewpoint is represented by Information Objects and Information actions and the data states.

Overview

Observation and measurement stations are established where specific requirements are fulfilled and thus there might exist a *specification of investigation design.*

The measurement is done with defined devices, arranged in defined geometries and according to defined setups. All those characteristics are kept together in the *measurement description*, which is planned to be stored in *metadata* and/or in a *data provenance* description.

The measurements at the *observation stations* produce *measurement results (L0 Data)* as they come out of the sensors*.* Usually this data is not calibrated and expressed in non-physical units, e.g. in milivolts. This raw data is not published.

For further steps those data are *persisted (stored for a longer period).*

A lot of *data handlings* are carried out with persistent data, changing the state. In ICOS different levels of data are defined: L0, NRT, L1, L2 and L3. In the information viewpoint of the RM different data states are specified according to the action which has been applied. When we adapt the Reference Model to the data levels of ICOS we have to consider that the actions can be applied in different orders and that they can be performed on different data states. Flagging the actions applied to the data it is possible to describe the different data levels as you can see in the following cross-table. In the data lifecycle (2.2.2) and the tables of chapters 2.2.3 and 2.2.4 the actual mapping is made evident.

	First automated quality check	Conversion to proper unit (pre-processed)	Manual QAQC	Gaps identified (quality flags)	Gaps filled (processed)	Averaged data	Metadata stored	Identifier	Provenance stored	User tracked	Backup	Published	Post-processed (post)

Measurement results:													
L0								p					
Persistent data:								p					
NRT	x	x					p	p	p	x	x/TC x/CP	x/TC	
L1	x	x	x	x			p	p	p	x	x/CP	x/TC	
L2	x	x	x	x	x	x	p	p	p	x	x/CP	x/CP	
L3	x	x	x	x	x	x	p	p	p	x	x/CP	x/CP	x

x… applied

p… planned to be applied

NRT (Near Real Time) data are stored at the Thematic Center (TC) pass a first-level automated quality check, a first processing to convert units from Voltage [mV] to Concentration [ppm] and, after a minimum time delay, are ready to be downloaded by “special” users (researchers) interested in very fast access.

L1 are stored at the Thematic Center data have undergone automated quality checks and checked by the Prinicipal Investigator of the observation station (“manual” QAQC processing). This process will likely result in “gaps”, where some parameters are given a “missing value” to indicate the data are unreliable (associating quality flags). L1 data are published at the Thematic Center and therefore are ready to be downloaded by selected researches.

L2 data are gap-filled data, where missing values are replaced with interpolated or otherwise modelled data (based e.g. on functions of meteorological parameters). They area consolidated and averaged data – at half-hourly or hourly frequency.

L3 data are elaborated data products, also called post-processed data. These would typically be data sets containing the outcome of different types of modeling - such as inverse modeling (using GHG concentration data from ICOS as part of the inputs) that gives "maps" showing sources and sinks of greenhouse gases distributed in time and space, or fluxes and other parameters calculated with ecosystem or vegetation models (using ICOS meteo, solar radiation and GHG gas data from specific ecosystem sites as input and/or as validation). Also other types of L3 products are possible, for example based on combining remote sensing data with ICOS data.

NRT, L1, L2, and L3 data are archived (backup) at data storages at the Carbon Portal (CP). NRT and L1 data are published at the TC, while L2 and L3 data are published at the CP. The usage of all levels of the persisted data are tracked by the CP, as the user interested in data is asked to register (at least a valid e-mail is required, plus probably indication of intended use). Persistent identifiers are planned to be assigned at each data level. Metadata at all levels are planned to be stored to keep track of a lot of information in order for the data sets to be useable and interpretable - ranging from info about the observations themselves (instruments, calibrations, potential issues), about the data processing (quality level, gapfilling method, flux evaluation methodology, date & time of TC processing) and of course about the datasets (revision, author, contact person, doi etc.). At the moment the metadata standard to be followed is not fixed. Published data will be ready to be queried via the CP providing a searching metadata service using a metadata repository.

By experience we know, that the *meaning* of the data, *the conceptual model* behind the data may change over time, as science proceeds. If those data shall be *processed together* they have to be mapped to a *common conceptual model.* We suggest therefore to build a local conceptual model for each Thematic Center and for the ICOS Carbon Portal which will then be used together for performing a mapping.

The first illustration gives an overview of the life cycle of ICOS data, seen from the information viewpoint. It reflects the state of ICOS, as it is already (more or less) implemented and representable by the current information viewpoint of the ENVRI RM.

The usage tracking action and all relevant information objects needed for this process are missing so far, as the important information for the user about the data genesis can also be gained by data provenance tracking without the need of storing information about the user and his intentions. As the ICOS head office is interested to introduce authorization and authentication to the ICOS portal enabling usage tracking, the necessary elements should be added to the data life cycle illustration in the reference model. Not yet implemented but planned are the addition of persistent unique identifiers, the metadata collection and the metadata query to support data search and discovery. The second illustration adds the information needed (in green shades) for usage tracking and for metadata to complete the picture of the planned Carbon Portal.

Data lifecycle seen from the ENVRI reference model

We provide the analysis of ICOS data lifecycle in Figure 3, which are instantiation of the Dynamic Schemata specified in the ENVRI Reference Model Information Viewpoint.

Figure 3: Analysis of ICOS Data Lifecycle from Reference Model Information Viewpoint

Information Objects

Table 3: Mapping of ICOS data object instances to the RM Information Objects

ENVRI RM IV_Objects	ICOS instance	existing/planned
Specification of investigation design	Specification of site requirements	existing
Measurement description	Specification of ICOS TC measurement or observation	existing
Measurement result	L0 data	existing

Persistent data, data state: raw	NRT data	existing
Persistent data, data state: QA assessed	L1	existing
Persistent data, data state: finally reviewed	L2	existing
Persistent data, data state: published	Published at TC	existing
Persistent data, data state: published	Published at CP	planned
Backup	ICOS RI Archive	existing
QA notation	Specification for automated quality check	existing
QA notation	Flag for gaps	existing
Unique identifier	PID?	planned
Data Provenance	Provenance information	planned
Metadata: state: raw	raw metadata	planned
Metadata: registered	Registered metadata	planned
Metadata: published	Published metadata	planned
Metadata catalogue	Metadata registry	planned
not yet implemented	Usage statistics	planned

Information Actions

Table 4: Mapping of ICOS data object instances to the RM Information Action Types

ENVRI RM IV_Action Types	**ICOS instance**	**existing/planned**
specify investigation design	specify site requirements	existing
specify measurement or observation	specify specific TC measurement or observation	existing
perform measurement or observation	perform ICOS measurement or observation	existing
store data	store ICOS data	existing
check quality	automated quality checking	existing
check quality	manual quality checking	existing
carry out backup	archive ICOS data	existing
publish data	publish data at TC	existing
publish data	publish data at CP	planned
process data	preprocessing (conversion)	existing
process data	building averages	existing
process data	gapfilling	existing
process data	Postprocess	existing
assign unique identifier	Assign PIDs?	planned
add metadata	add ICOS metadata	planned
register metadata	register ICOS metadata	planned
publish metadata	publish ICOS metadata	planned
query data	query ICOS data	Planned
do data mining	do data mining	planned

Annotate action	Annotate action	planned
query provenance	query provenance	planned
not yet implemented	track usage	planned

Analysis of ICOS Research Infrastructure from *Computational Viewpoint*

The Integrated Carbon Observation System (ICOS; http://www.icos-infrastructure.eu/) is a distributed research infrastructure involving a number of key facilities and services:

An overview of the proposed data-flow in ICOS (April 2014).

From the Computational Viewpoint, each of the core ICOS facilities (principally the thematic centres and the Carbon Portal) is responsible for providing a number of infrastructure functions. Within the infrastructure as a whole, there exist a number of interactions within and between these facilities that must be modelled; modelling these interactions will (a) ensure that key use-cases have been accounted for and (b) provide a basis for component-wise comparison with other related infrastructure projects.

The deconstruction of the ICOS research infrastructure adheres to the terminology defined How to read the Model (Computational Viewpoint); the methodology of the (initial) modelling of the ICOS infrastructure is based on principles defined How to use the Model (Computational Viewpoint).

The current version of the ENVRI Reference Model is deemed to be a minimal model, in that it concentrates on the critical data pipelines between parts of an infrastructure. As a result, some aspects of the ICOS specification (such as the data visualisation capabilities of the Carbon Portal) are *not* properly accounted for in the descriptions below, but could be added later.

Core computational objects

The first observation to be made about the ICOS computational infrastructure is that there are multiple curation sites; each of the thematic centres stores data, as does the Carbon Portal (via its backend services). This means that there are multiple instances of CV Data Curation, with different associated interaction models.

A catalogue of ICOS computational objects by site.

Several important computational objects required by each major 'site' in ICOS are described below along with the bindings for which interaction models will need to be specified. *This is a provisional survey, subject to further information about and refinement of the roles of each of the thematic centres and the Carbon Portal.*

Thematic Centres

All three thematic centres (Atmospheric, Ecological and Ocean) possess instances of the same computational objects regardless of differences in their instrument networks and how they acquire observations from those networks. There may however be a different number of instances of a given object, and their binding behaviours may be different (for example, the ATC and ETC may have different numbers of instrument controllers, and may use different interaction models for a *configure instrument* binding).

The thematic centres are responsible for producing almost all scientific data within ICOS, and are able to operate autonomously from the ICOS Carbon Portal. However for the purposes of identifying key computational services that must be hosted by various sites, the assumption is that all external requests are filtered through the Carbon Portal; thus there may be instances of services (particularly related to data access) for which computational objects are not listed below because it is assumed that their function will be carried out by the Carbon Portal for all interactions originating from the Portal.

CV Data Acquisition

Field laboratory : Each thematic centre provides an environment for deploying and calibrating instruments in their respective measurement networks; this is done mostly manually by scientists and technicians in the field however, with most interactions with instruments being physical rather than virtual.

> Bindings: calibrate [atmospheric/ecosystem/ocean] instrument, update [atmospheric/ecosystem/ocean] registry.

Acquisition service : Each thematic centre will have one or more acquisition services to handle the acquisition of data from measurement networks and ensure that L0 data is recorded within data stores within the centre; as with the field laboratory, some of this service's functionality may be carried out manually.

> Bindings: configure [atmospheric/ecosystem/ocean] controller, prepare [atmospheric/ecosystem/ocean] transfer, update [atmospheric/ecosystem/ocean] registry.

CV Data Curation

Catalogue service : Each thematic centre will catalogue its L0, L1 and L2 data and store associated metadata. A catalogue service will be required at each thematic centre to catalogue the centre's complete data corpus and preserve the link between metadata and data.

> Bindings: archive [L0/L1/L2] data, collect L0 [atmospheric/ecosystem/ocean] data, derive [L1/L2] [atmospheric/ecosystem/ocean] data, export [L0/L1] [atmospheric/ecosystem/ocean] data, query [atmospheric/ecosystem/ocean] data, query [atmospheric/ecosystem/ocean] resource.

Data store controller : Each thematic centre stores L0, L1 and L2 data as well as associated metadata. Data store controllers will be necessary for all types of data stored, with different interaction models for data entry and access for different types of data store.

> Bindings: archive [L0/L1/L2] data, collect L0 [atmospheric/ecosystem/ocean] data, derive [L1/L2] [atmospheric/ecosystem/ocean] data, export [L0/L1] [atmospheric/ecosystem/ocean] data, process [L0/L1] [atmospheric/ecosystem/ocean] data, query [atmospheric/ecosystem/ocean] resource.

Data transfer service : Each thematic centre is responsible for serving L0 and L1 data on request, as well as sending any L2 data generated to the Carbon Portal for preservation. Each thematic centre may provide multiple data transfer services to handle different classes of data request, or have one data transfer service to manage all transfers.

> Bindings: prepare [L0/L1/L2] [atmospheric/ecosystem/ocean] data archival, prepare L0 [atmospheric/ecosystem/ocean] data collection, prepare [L0/L1] [atmospheric/ecosystem/ocean] data export, prepare [L0/L1] [atmospheric/ecosystem/ocean] data staging, prepare [L1/L2] [atmospheric/ecosystem/ocean] result transfer.

CV Data Processing

Coordination service : Each thematic centre is capable of deriving L1 data from L0 data, and L2 data from L1 data. The role of the coordination service is to manage the processing of data by arranging the staging of data onto processing resources and the reclamation of results; each thematic centre should have a coordination service to coordinate the derivation of new datasets.

> Bindings: coordinate [atmospheric/ecosystem/ocean] process, prepare [L0/L1] [atmospheric/ecosystem/ocean] data staging, prepare [L1/L2] [atmospheric/ecosystem/ocean] result transfer, request [atmospheric/ecosystem/ocean] process.

Experiment laboratory : Each thematic centre provides an environment for systematic processing of L0 data in order to produce L1 data as well as L2 data from L1 data; an experiment laboratory encapsulates the functions required to describe and request computational processes that can be used to derive higher-level datasets from lower-level ones. As such, each thematic centre can be stated to possess at least one experiment laboratory.

> Binding: request [atmospheric/ecosystem/ocean] process.

Process controller : Each thematic centre performs systematic processing of acquired data. A process controller provides an interface to a process, allowing data to be staged, processed, and the results stored. There should be process controllers present at each thematic centre representing their respective data derivation processes.

> Bindings: coordinate [atmospheric/ecosystem/ocean] process, derive [L1/L2] [atmospheric/ecosystem/ocean] data, process [L0/L1] [atmospheric/ecosystem/ocean] data.

CV Data Use

PID service : ICOS will require a mechanism by which to assign persistent identifiers to datasets. If these PIDs are to be globally distinguishable, then it will likely be necessary to use an external PID handling service. If the PIDs need only be distinguishable within the ICOS context, then it may be sufficient to host a PID service in the Carbon Portal, or it may be expedient to host PID services at each thematic centre.

> Bindings: collect L0 [atmospheric/ecosystem/ocean] data, derive [L1/L2] [atmospheric/ecosystem/ocean] data.

Measurement Station Networks

Each theme has a network of measurement stations associated with it that provides their respective thematic centre with L0 data.

Instrument controller : An instrument is considered *computationally* to be any source of observation data deployed 'in the field'. An instrument controller object encapsulates the computational functions required to interact with an instrument and acquire data from it. In ICOS, instruments may correspond to measurement stations rather than individual sensors installed within stations if computationally the stations act as a single interactive entity. The chosen fidelity for instruments need not be the same for all themes however. Regardless, each theme will have a number of instrument controllers that interact with services in the respective thematic centre.

> Bindings: calibrate [atmospheric/ecosystem/ocean] instrument, configure [atmospheric/ecosystem/ocean] controller, collect L0 [atmospheric/ecosystem/ocean] data.

Carbon Portal

The Carbon Portal provides a gateway into the ICOS research infrastructure and provides access to L2 and L3 data along with visualisation services.

CV Data Curation

Annotation service : The role of an annotation service is to provide a mechanism to add or edit the metadata associated with a dataset, as well as add generic user annotations to data if supported. The Carbon Portal may provide annotation services for L2 and L3 data for example.

> Bindings: annotate [L2/L3] data, annotate metadata, update [L2/L3] catalogues, update [L2/L3] records, update metadata records.

Catalogue service : The Carbon Portal will require a catalogue service to catalogue all L2 and L3 data and their associated metadata. Such a service will likely have access to catalogues of L0 and L1 data stored at all three thematic centres as well. Usage statistics and data analyses preserved on-site may also require cataloguing, depending on their complexity and whether or not they are stored permanently.

> Bindings: archive [L0/L1/L2/L3] [atmospheric/ecosystem/ocean] data, export [L2/L3] data, import L3 data, query CP data, query CP resource.

Data store controller : The Carbon Portal stores L2 and L3 data as well as associated metadata and usage statistics. Data store controllers will be necessary for all types of data stored, with different interaction models for data entry and access for different types of data store.

> Bindings: archive [L0/L1/L2] [atmospheric/ecosystem/ocean] data, export [L2/L3] data, import L3 data, query resource.

Data transfer service : The Carbon Portal must be able to provide L2 and L3 datasets on request, as well as be able to import certain external datasets into the infrastructure. Data is also transferred between thematic centres and the Carbon Portal (though whether the data transfer service at the Carbon Portal or the transfer service at the thematic centres handles this task is a matter or architectural convenience).

> Bindings: prepare [L0/L1/L2] [atmospheric/ecosystem/ocean] data archival, prepare [L2/L3] data export, prepare L3 data import.

CV Data Access

Data broker : One of the Carbon Portal's primary roles is to provide access to a range of scientific datasets. The role of the data broker is validate data requests and identify where data is stored, as well as authorise any resulting data transfers. The Carbon Portal may host a number of data brokers for different kinds of data request or query, or may integrate them all into one service.

> Bindings: annotate [L2/L3] data, perform data query, prepare [L0/L1/L2] [atmospheric/ecosystem/ocean] data archival, prepare [L0/L1] [atmospheric/ecosystem/ocean] data export, query [atmospheric/ecosystem/ocean] data, request [L0/L1] data export, request [L2/L3] data export, request L3 data import, query CP data.

CV Data Use

PID service : ICOS will require a mechanism by which to assign persistent identifiers to datasets. If these PIDs are to be globally distinguishable, then it will likely be necessary to use an external PID handling service. If the PIDs need only be distinguishable within the ICOS context, then it may be sufficient to host a PID service in the Carbon Portal, or it may be expedient to host PID services at each thematic centre.

> Binding: import L3 data.

Science gateway : A science gateway is a service offering access to the rest of the infrastructure. The Carbon Portal exists to provide such a gateway.

> *Science gateways are capable of creating new virtual laboratories for users.*

Security service : A security service is required to validate user requests and verify identity. The level of security (or identity management, if preferred) required depends on the scope and complexity of services offered by the Carbon Portal in practice.

- Binding: authorise action.

Virtual laboratory : Another primary role of the Carbon Portal is to provide a virtual research environment for investigating ICOS data. This environment may be simply a means to download, upload and visualise datasets, or may provide more elaborate services (such as user accounts, processing privileges for virtual organisations, *etc*.). A virtual laboratory object encapsulates the services provided to a given user or set of users in a given context (such as a browser session).

- Bindings: authorise action, perform data query, request [L0/L1] data export, request [L2/L3] data export, request L3 data import.

Core bindings

The following bindings require interaction models to be defined within the ICOS infrastructure to be compliant with the ENVRI Reference Model. Many of the binding descriptions below describe multiple similar but distinct bindings (for example **archive [L0/L1/L2] [atmospheric/ecosystem/ocean] data** describes 9 different bindings in total) – in principle, an interaction model is required for *each* individual binding (though many or all of those models may be nearly identical). Most bindings are primitive bindings between two instances of the computational objects described above, but some compound bindings are defined as well; these compound bindings have links to dedicated subsections below. Compound bindings generally combine multiple primitive bindings together to create a single interaction; a single unified interaction model can be defined for each compound binding, or individual models can be defined for primitive sub-bindings and composed as deemed most appropriate.

In ICOS, a significant number of bindings are to different thematic centres or involve different levels of dataset (or both). Because different thematic centres may organise themselves differently, and different data policies may apply to different levels of data, different interaction models may apply to different cases of what is otherwise, to the Reference Model, the same abstract interaction; hence the proliferation of nearly-identical types of binding.

archive [L0/L1/L2] [atmospheric/ecosystem/ocean] data : Used to export [L0/L1/L2] datasets from the [ATC/ETC/OTC] to the Carbon Portal's own archives. See archive_data.

authorise action : Used to retrieve authentication tokens required to authorise a variety of actions across the infrastructure. A user should invoke authorise action before almost any other action.

- **client** : Any virtual laboratory on behalf of an agent wishes to interact with the Carbon Portal.
- **interface** : authorise action
- **server** : The Carbon Portal security service.

annotate [L2/L3] data : Used to edit or annotate [L2/L3] data held by the Carbon Portal.

- **client** : Any Carbon Portal data broker acting on behalf of an authorised agent.
- **interface** : annotate data
- **server** : The annotation service provided by the Carbon Portal for the agent that wishes to perform the edit/annotation.

annotate metadata : Used to edit metadata held within the Carbon Portal metadata repository.

- **client** : Any Carbon Portal data broker acting on behalf of an authorised agent.
- **interface** : annotate data
- **server** : The annotation service provided by the Carbon Portal for the agent that wishes to perform the edit/annotation.

calibrate [atmospheric/ecosystem/ocean] instrument : Used to monitor and calibrate instruments in the [atmospheric/ecosystem/ocean] theme's measurement station network.

- **client** : Any field laboratory in the [ATC/ETC/OTC].
- **interface** : calibrate instrument
- **server** : Any instrument controller in the [atmospheric/ecosystem/ocean] theme measurement station network.

collect L0 [atmospheric/ecosystem/ocean] data : Used to retrieve L0 data from instruments and store them in the [ATC/ETC/OTC]. See collect_L0_.

configure [atmospheric/ecosystem/ocean] controller : Used to control how and when data an instrument sends data to the [ATC/ETC/OTC].

- **client** : The acquisition service in the [ATC/ETC/OTC] associated with the instrument(s) to be configured.
- **interface** : configure controller
- **server** : The instrument controller associated with the instrument(s) to be configured.

coordinate [atmospheric/ecosystem/ocean] process : Used to configure, run and monitor a processing task on a processing resource.

- **client** : The coordination service in the [ATC/ETC/OTC] responsible for managing the process in question.
- **interface** : coordinate process
- **server** : The process controller associated with the process being executed.

derive [L1/L2] [atmospheric/ecosystem/ocean] data : Used to derive higher-level data from datasets held by the [ATC/ETC/OTC]. See derive_TC_data.

export [L0/L1] [atmospheric/ecosystem/ocean] data : Used to export [L0/L1] datasets from the [ATC/ETC/OTC] to an external resource. See export_TC_data.

export [L2/L3] data : Used to export [L2/L3] datasets from the Carbon Portal to an external resource. See export_CP_data.

import L3 data : Used to upload L3 datasets from an external resource into the Carbon Portal. See import_.

perform data query : Used to request that a data broker perform a query over the aggregate data held by the ICOS infrastructure.

- **client** : The virtual laboratory used by the agent making the request.
- **interface** : data request
- **server** : The Carbon Portal data broker responsible for brokering queries over Carbon Portal data.

prepare [L0/L1/L2] [atmospheric/ecosystem/ocean] data archival : Used to initiate the transfer of [L0/L1/L2] data stored in the [ATC/ETC/OTC] to the archives of the Carbon Portal.

- **client** : The Carbon Portal data broker responsible for replicating [ATC/ETC/OTC] [L0/L1/L2] data.
- **interface** : prepare data transfer
- **server** : The data transfer service responsible for managing the transfer of [L0/L1/L2] datasets from the [ATC/ETC/OTC] to the Carbon Portal.

prepare L0 [atmospheric/ecosystem/ocean] data collection : Used to initiate the transfer of L0 data from instruments in the [atmospheric/ecosystem/ocean] measurement station network to the [ATC/ETC/OTC].

- **client** : The acquisition service in the [ATC/ETC/OTC] associated with the instrument(s) providing data.
- **interface** : prepare data transfer
- **server** : The data transfer service responsible for managing the collection of L0 data for the [ATC/ETC/OTC].

prepare [L0/L1] [atmospheric/ecosystem/ocean] data export : Used to initiate the export of [L0/L1] data stored in the [ATC/ETC/OTC] to an external resource.

- **client** : The Carbon Portal data broker responsible for requesting [L0/L1] data stored in the [ATC/ETC/OTC].
- **interface** : prepare data transfer
- **server** : The data transfer service responsible for managing the export of [L0/L1] data from the [ATC/ETC/OTC].

prepare [L0/L1/L2] [atmospheric/ecosystem/ocean] data staging : Used to initiate the staging of [L0/L1/L2] data stored in the [ATC/ETC/OTC] into a suitable context for processing.

- **client** : The coordination service responsible for deriving higher-level data from [L0/L1/L2] datasets held in the [ATC/ETC/OTC].
- **interface** : prepare data transfer
- **server** : The data transfer service responsible for managing the staging of [L0/L1/L2] datasets held in the [ATC/ETC/OTC].

prepare [L1/L2] result transfer : Used to initiate the retrieval of new [L1/L2] processed data from processing facilities to be stored in the [ATC/ETC/OTC].

- **client** : The coordination service responsible for deriving [L1/L2] data from lower-level datasets held in the [ATC/ETC/OTC].
- **interface** : prepare data transfer
- **server** : The data transfer service responsible for managing the retrieval of [L1/L2] datasets to be held in the [ATC/ETC/OTC].

prepare [L2/L3] data export : Used to initiate the export of [L2/L3] data stored in the Carbon Portal's archives to an external resource.

- **client** : The Carbon Portal data broker responsible for fielding requests for [L2/L3] data.
- **interface** : prepare data transfer
- **server** : The data transfer service responsible for managing the export of [L2/L3] data from the Carbon Portal.

prepare L3 data import : Used to initiate the upload of L3 data from an external resource to the Carbon Portal's archives.

- **client** : The Carbon Portal data broker responsible for fielding requests to upload L3 data.
- **interface** : prepare data transfer
- **server** : The data transfer service responsible for managing the import of L3 data into the Carbon Portal.

process [L0/L1] [atmospheric/ecosystem/ocean] data : Used to stage scientific data held by the [ATC/ETC/OTC] for analysis and processing. See process_TC_data.

query [atmospheric/ecosystem/ocean] data : Used by the Carbon Portal to query the aggregate data held by the [ATC/ETC/OTC].

- **client** : Any Carbon Portal data broker.
- **interface** : query data
- **server** : The [ATC/ETC/OTC] catalogue service.

query [atmospheric/ecosystem/ocean] resource : Used by the catalogue service within the [ATC/ETC/OTC] to query the data held within a specific data store within the [ATC/ETC/OTC].

- **client** : The [ATC/ETC/OTC] catalogue service.
- **interface** : query resource
- **server** : The data store controller within the [ATC/ETC/OTC] that holds the desired data to be queried.

query CP data : Used by the Carbon Portal to query the aggregate data held within its own archives.

- **client** : Any Carbon Portal data broker.
- **interface** : query data
- **server** : The Carbon Portal catalogue service.

request [atmospheric/ecosystem/ocean] process : Used to request that a particular data processing task be executed within the [ATC/ETC/OTC].

- **client** : The experiment laboratory providing the environment for data processing within the [ATC/ETC/OTC].
- **interface** : process request
- **server** : Any coordination service present within the [ATC/ETC/OTC] capable of executing the given process.

request [L0/L1] data export : Used to request the export of an [L0/L1] dataset from the relevant thematic centre (performed via the Carbon Portal; may be implemented as a manual request).

- **client** : The virtual laboratory used by the agent making the request.
- **interface** : data request
- **server** : The Carbon Portal data broker responsible for brokering access to [L0/L1] data.

request [L2/L3] data export : Used to request the export of an [L2/L3] dataset from the Carbon Portal's archives.

- **client** : The virtual laboratory used by the agent making the request.
- **interface** : data request
- **server** : The Carbon Portal data broker responsible for brokering access to [L2/L3] data.

request L3 data import : Used to request permission to upload L3 synthesis data to the Carbon Portal's archives.

- **client** : The virtual laboratory used by the agent making the request.
- **interface** : data request
- **server** : The Carbon Portal data broker responsible for brokering uploads of syntheses to the Carbon Portal.

update [L2/L3] catalogues : Used to perform annotation updates on catalogues managed by the Carbon Portal catalogue service (systematic updates are handled as part of data transfers such as for archive_data).

- **client** : The Carbon Portal annotation service.
- **interface** : update catalogues
- **server** : The catalogue service used by the Carbon Portal to manage [L2/L3] data catalogues.

update [L2/L3] records : Used to perform annotation updates to data recorded in an [L2/L3] data store within the Carbon Portal's archives (systematic updates are handled as part of data transfers such as for archive_data).

- **client** : The Carbon Portal annotation service.
- **interface** : update records
- **server** : The [L2/L3] data store controller used by the Carbon Portal to control the data store containing the data to be annotated.

update metadata records : Used to perform updates to metadata recorded in the Carbon Portal's metadata repository.

- **client** : The Carbon Portal annotation service.
- **interface** : update records
- **server** : The data store controller used by the Carbon Portal to control its metadata repository.

update [atmospheric/ecosystem/ocean] registry : Used to register and unregister instruments deployed in the [atmospheric/ecosystem/ocean] theme's measurement station network.

- **client** : Any field laboratory in the [ATC/ETC/OTC].
- **interface** : update registry
- **server** : The acquisition service in the [ATC/ETC/OTC] with which the instrument(s) involved are registered / to be registered.

Compound Bindings

Compound bindings are used to bind three or more computational objects by means of an intermediary binding object. The binding object is responsible for coordinating interaction between the bound computational objects. Those core bindings described above that have been deemed to be compound bindings are described in more detail here. Note that these descriptions merely explicate the set of objects being bound in each case and their purpose; just as for the primitive bindings, a compliant infrastructure must define interaction models for each compound binding.

The compound binding descriptions below are deliberately pedantic in how they distribute oversight, control and data-flow between objects. However in practice, many of the separate objects may be collapsed together to simplify the interaction supported; for example the staging of data from a permanent data store to a processing platform, with the results then moved back into another data store, might simplify to just performing data processing in-situ within one data store without any data movement.

archive [L0/L1/L2] data

The archival of L0, L1 and L2 data produced by any of the thematic centres by the Carbon Portal binds a thematic data store to a designated Carbon Portal data store. The binding occurs at two levels: at the operational level, data is requested via the thematic data store controller's *retrieve data* interface and the CP data store's internal records are updated via its controller's *update records* interface; at the data streaming level, a data channel is set up to transfer curated data. Additional metadata is retrieved by the relevant thematic catalogue service whilst the CP's catalogue service updates the its catalogue of L0, L1 or L2 datasets.

Either the relevant thematic centre's data transfer service or the CP's data transfer service creates the *data transporter* binding object necessary to carry out the interaction. All datasets of all levels should be archived for data preservation purposes (via a prepare_data_archival binding), even though only L2 and L3 data is directly served by the Carbon Portal.

collect L0 data

The collection of L0 data from any of the thematic measurement station networks binds a instrument controller to a data store within the corresponding thematic centre. The binding occurs at two levels: at the operational level, data is requested via the instrument controller's *retrieve data* interface and the data store's internal records are updated via its controller's *update records* interface; at the data streaming level, a data channel is set up to deliver raw data for curation. A persistent identifier is acquired from the thematic PID service and the thematic catalogue service updates the corresponding thematic centre's catalogue of L0 datasets.

The relevant thematic centre's data transfer service creates the *data transporter* binding object necessary to carry out the interaction. The data transfer service will only set up a data channel between an instrument and a data store if the relevant acquisition service requests it (via a prepare_raw_data_transfer binding).

derive [L1/L2] data

The derivation of L1 or L2 data within any of the thematic centres binds a process controller to an L1 or L2 data store respectively. The binding occurs at two levels: at the operational level, derived data is requested via the process controller's *retrieve data* interface and the data store's internal records are updated via its controller's *update records* interface; at the data streaming level, a data channel is set up to deliver the derived data for curation. A persistent identifier is acquired from the thematic PID service and the thematic catalogue service updates the corresponding thematic centre's catalogue of L1 or L2 datasets.

The relevant thematic centre's data transfer service creates the *data transporter* binding object necessary to carry out the interaction. The data transfer service will only set up a data channel between a process controller and a data store if the relevant coordination service requests it (via a prepare_data_staging binding). The derivation of any level of data is precluded by the process_L0_data. Note that if data has been processed within the data store it resides, then this binding can be implemented trivially.

export [L0/L1] data

The export of L0 or L1 data from any of the thematic centres binds a thematic data store to a designated data store outside of the ICOS infrastructure. The binding occurs at two levels: at the operational level, data is requested via the data store controller's *retrieve data* interface; at the data streaming level, a data channel is set up to export curated data. Additional metadata is retrieved by the thematic catalogue service.

The relevant thematic centre's data transfer service creates the *data transporter* binding object necessary to carry out the interaction. The data transfer service will only export L0 or L1 data on request via the Carbon Portal (via a prepare_TC_data_export binding or via direct request to the thematic centre; but this is auxiliary to ICOS).

export [L2/L3] data

The export of L2 or L3 data from the Carbon Portal binds a data store in the Carbon Portal's archives to a designated data store outside of the ICOS infrastructure. The binding occurs at two levels: at the operational level, data is requested via the archive data store controller's *retrieve data* interface; at the data streaming level, a data channel is set up to export curated data. Additional metadata is retrieved by the Carbon Portal catalogue service.

The Carbon Portal's data transfer service creates the *data transporter* binding object necessary to carry out the interaction. The data transfer service will only export L2 or L3 data upon a valid request being made (via a prepare_CP_data_export binding).

import L3 data

The upload of externally-produced L3 syntheses into the ICOS infrastructure binds an external resource to a data store within the Carbon Portal's archives. The binding occurs at two levels: at the operational level, the Carbon Portal data store's internal records are updated via its controller's *update records* interface; at the data streaming level, a data channel is set up to import data for curation. A persistent identifier is acquired from the Carbon Portal's PID service and the Carbon Portal's catalogue service updates the Carbon Portal's catalogue of L3 syntheses.

The Carbon Portal's data transfer service creates the *data transporter* binding object necessary to carry out the interaction. The data transfer service will only set up a data channel between an external resource and a data store if a valid upload request is made (via a prepare_data_import binding).

process [L0/L1] data

The processing of L0 or L1 data within any of the thematic centres binds a L0 or L1 data store respectively to a process controller. The binding occurs at two levels: at the operational level, data is requested via the data store controller's *retrieve data* interface and the process controller's internal records are updated via its controller's *update records* interface; at the data streaming level, a data channel is set up to stage the L0 or L1 data.

The relevant thematic centre's data transfer service creates the *data stager* binding object necessary to carry out the interaction. The data transfer service will only set up a data channel between a data store and a processing context if the relevant coordination service requests it (via a prepare_result_transfer binding). The processing of any level of data precludes the derive_L1_data. Note that if data can be processed within the data store it resides, then this binding can be implemented trivially.

Bibliography

1. W. Los, "Introduction to ENVRI and the workshop objectives," in *ENVRI Frascati Meeting 5-7 Feb 2013*, Presentation. Frascati, Italy, 2013.
2. "Global Change: Towards global research infrastructures," *European Commission, Directorate-General For Research and Innovation*, 2012.
3. S. Sorvari. "Envrionmental reseach in harmony," *International Innovation - Disseminating, science research and technology*. Dec. 2012 Page 28, 2012. Available: **http://www.research-europe.com/magazine/ENVIRONMENT/2012-15/index.html**
4. ISO/IEC, "ISO/IEC 10746-1: Information technology--Open Distributed Processing--Reference model: Overview," *ISO/IEC Standard*, 1998.
5. ISO/IEC, "ISO/IEC 10746-2: Information technology--Open Distributed Processing--Reference model: Foundations," *ISO/IEC Standard*, 2009.
6. ISO/IEC, "ISO/IEC 10746-3: Information technology--Open Distributed Processing--Reference model: Architecture," *ISO/IEC Standard*, 2009.
7. ISO/IEC, "ISO/IEC 10746-4: Information technology--Open Distributed Processing--Reference model: Architecture Semantics," *ISO/IEC Standard*, 1998.
8. OASIS, "Reference Model for Service Oriented Architecture 1.0," *OASIS Standard*, 2006.
9. L. Candela, G. Athanasopoulos, D. Castelli, K. El Raheb, P. Innocenti, Y. Ioannidis, A. Katifori, A. Nika, G. Vullo, and S. Ross. "The Digital Library Reference Model", *DL.org,* 2011. **http://referencemodel.dlorg.eu/**
10. ISO/IEC, "Open System Interconnection (OSI), ISO/IEC 7498-1," *ISO/IEC Standard*, 1994.
11. CCSDS, "Reference Model for an Open Archival Information System (OAIS)," *CCSDS Standard*, 2012.
12. C. Atkinson, M. Gutheil, and K. Kiko, "On the Relationship of Ontologies and Models," *Lecture Notes in Informatics, Gesellschaft für Informatik, Bonn,* INI Proceedings, 1996.
13. D. C. Schmidt, "Model-Driven Engineering," *IEEE Computer* vol. 39, 2006.
14. N. F. Noy, and D. L. McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology," *Stanford Knowledge*

Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, 2001.

15. P. Tetlow, J. Z. Pan, D. Oberle, E. Wallace, M. Uschold, and E. Kendall, "Ontology Driven Architectures and Potential Uses of the Semantic Web in Systems and Software Engineering," *W3C Standard*, 2006.
16. SEKE, "International Conference on Software Engineering (SEKE 2005)".
17. VORTE, "International Workshop on Vocabularies, Ontologies and Rules for The Enterprise (VORTE 2005-2013)".
18. MDSW, "The Model-Driven Semantic Web Workshop (MDSW 2004)".
19. SWESE, "Workshop on Semantic Web Enabled Software Engineering (SWESE 2005-2007)".
20. ONTOSE, "Workshop on Ontology, Conceptualizations and Epistemology of Software and Systems Engineering (ONTOSE 2005-2009)".
21. WoMM, "Workshop on Meta-Modeling and Corresponding Tools (WoMM 2005)" .
22. I. Kwaaitaal, M. Hoogeveen, and T. V. D. Weide, "A Reference Model for the Impact of Standardisation on Multimedia Database Management Systems," *Computer Standards & Interfaces,* vol. 16, pp. 45-54, 1994.
23. OGC, "OGC Reference Model," *Open Geospatial Consortium*, OGC Standard, 2011.
24. T. Uslander, "Reference Model for the ORCHESTRA Architecture (RM-OA) V2,", *Open Geospatial Consortium*, OGC Standard, 2007.
25. V. Hernandez-Ernst, *et al.*, "LIFEWATCH. Deliverable 5.1.3: Data & Modelling Tool Structures -- Reference Model," *the EU LifeWatch consortium*, 2010.
26. D. Hollingsworth, "The Workflow Reference Model," *the Workflow Management Coalition*, 1995.
27. I. Mayk, and W. C. Regli, "Agent Systems Reference Model Release Version 1.0a," *US Army Communications and Electronics Command Research Development and Engineering Center (CERDEC)*, 2006.
28. E. H. Chi, and J. T. Riedl, "An Operator Interaction Framework for Visualization Systems," *Symposium on Information Visualization (InfoVis '98),* 1998.
29. E. H. Chi, "A Taxonomy of Visualisation Techniques using the Data State Reference Model," *Proceedings of the IEEE Symposium on Information Visualization 2000 (InfoVis'00),* 2000.
30. N. Koch, and M. Wirsing, "The Munich Reference Model for Adaptive Hypermedia Applications," *in 2nd International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, Proceedings. P. De Bra, P. Brusilovsky, and R. Conejo (eds.) LNCS 2347, ©Springer Verlag,* pp. 213-222, 2002.
31. OMG, "Data Distribution Service for Real-time Systems Version 1.2", *OMG Standard*, 2007.
32. OASIS, "Content Management Interoperability Services (CMIS) Version 1.0," *OASIS Stanard*, 2011.
33. A. Hardisty, "WP3 Progress and Issues for Full Plenary with SAB, Wednesday 6th Feb 2013," in *ENVRI Frascati Meeting, 5-7 Feb 2013*, ed. Frascati, Italy, 2013
34. R. Kahn and R. Wilensky, "A framework for distributed digital object services", *International Journal on Digital Libraries (2006) 6(2):115-123*, 2006.
35. A. Barros, M. Dumas and P. Oaks, "A Critical Overview of the Web Services Choreography Description Language (WS_CDL)", *BPTr ends*, Mar. 2005.
36. L. Candela. "Data Use - Virtual Research Environments". In K. Ashley, C. Bizer, L. Candela, D. Fergusson, A. Gionis, M. Heikkurinen, E. Laure, D. Lopez, C. Meghini, P. Pagano, M. Parsons, S. Viglas, D. Vitlacil, and G. Weikum, *(ed.) Technological & Organisational Aspects of a Global Research Data Infrastructure - A view from experts, GRDI2020, 91-98*, 2012,
37. P. F. Linington, Z. Milosevic, A. Tanaka, and A. Vallecillo, Ed., *Building Enterprise Systems with ODP*. CRC Press, 2012.
38. Oracle, *Oracle Information Architecture: An Architect's Guide to Big Data*, An Oracle White Pater in Enterprise Architecture, August 2012.
39. Tarasova, T., Argenti M., and Marx M., *Semantically-Enabled Environmental Data Discovery and Integration: demonstration using the Iceland Volcano Use Case*, To appear in proc. of the 4th Conference on Knowledge Engineering and Semantic Web (KESW), Saint-Petersburg, Russia, 2013.
40. OMG, "Unified Modeling LanguageTM (OMG UML), Superstructure Version 2.2" , *OMG Standard*, 2009
41. Regulation, Council. "No 723/2009 of 25 June 2009 on the Community legal framework for a European Research Infrastructure Consortium (ERIC)." *OJ L*206.8.8 (2009). **https://ec.europa.eu/research/infrastructures/pdf/eric_en.pdf**
42. ENVRI plus. (2016). D5.2 A definition of the ENVRI Reference Mode, Envri plus project Horizon 2020 Grant Agreement No 654182. **http://www.envriplus.eu/wp-content/uploads/2015/08/D5.2-A-definition-of-the-ENVRIplus-Reference-Model.pdf**