Lecture 9 – Metadata management

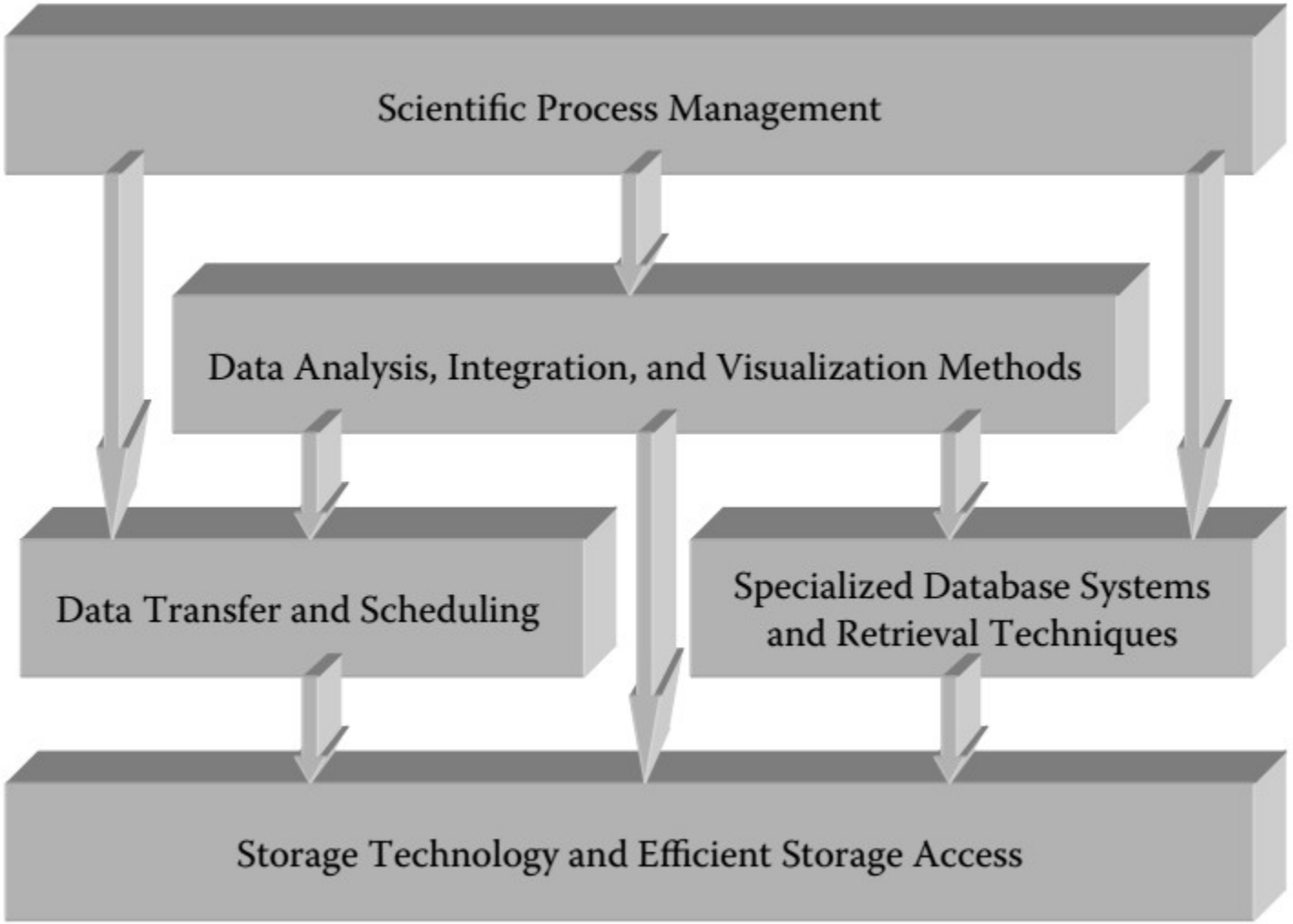
Open Data Management & the Cloud

(Data Science & Scientific Computing / UniTS – DMG)

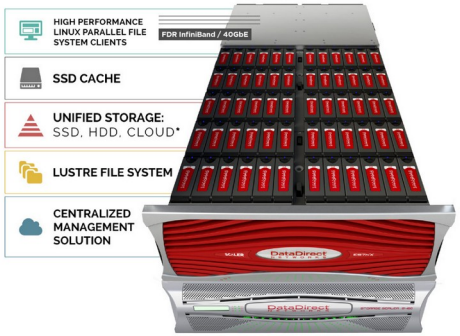
Scientific data management

- During the scientific exploration process, from the data generation phase to the data analysis phase, data management involves five main aspects
  - the efficient access to storage systems, in particular, parallel file systems, to write and read large volumes of data
  - A second aspect is the efficient data movement and management of storage spaces
  - techniques for automatically optimizing the physical organization of data, necessary for fast analysis
  - how to effectively perform complex data analysis and searches over large datasets
  - the automation of multistep scientific process workflows

Data management technologies



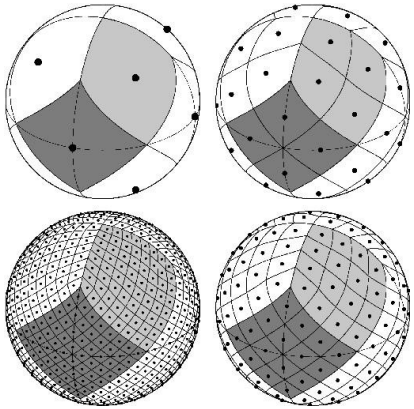
Data management technolgies



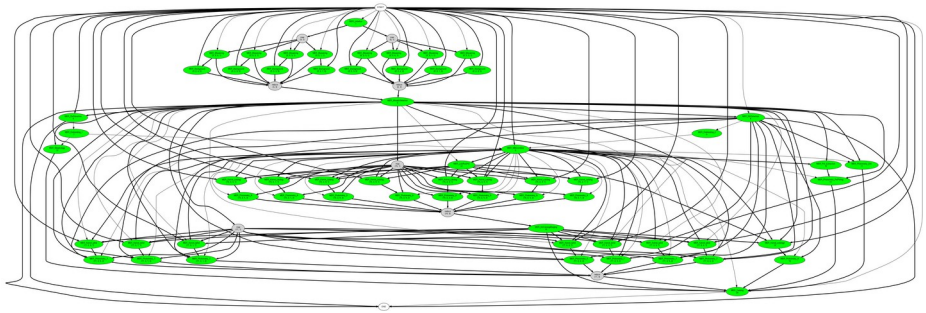
36 - 40

1.02PB-6.2PB HDD
926TB-4.76PB SSD

Up to 60GB/s / Rack



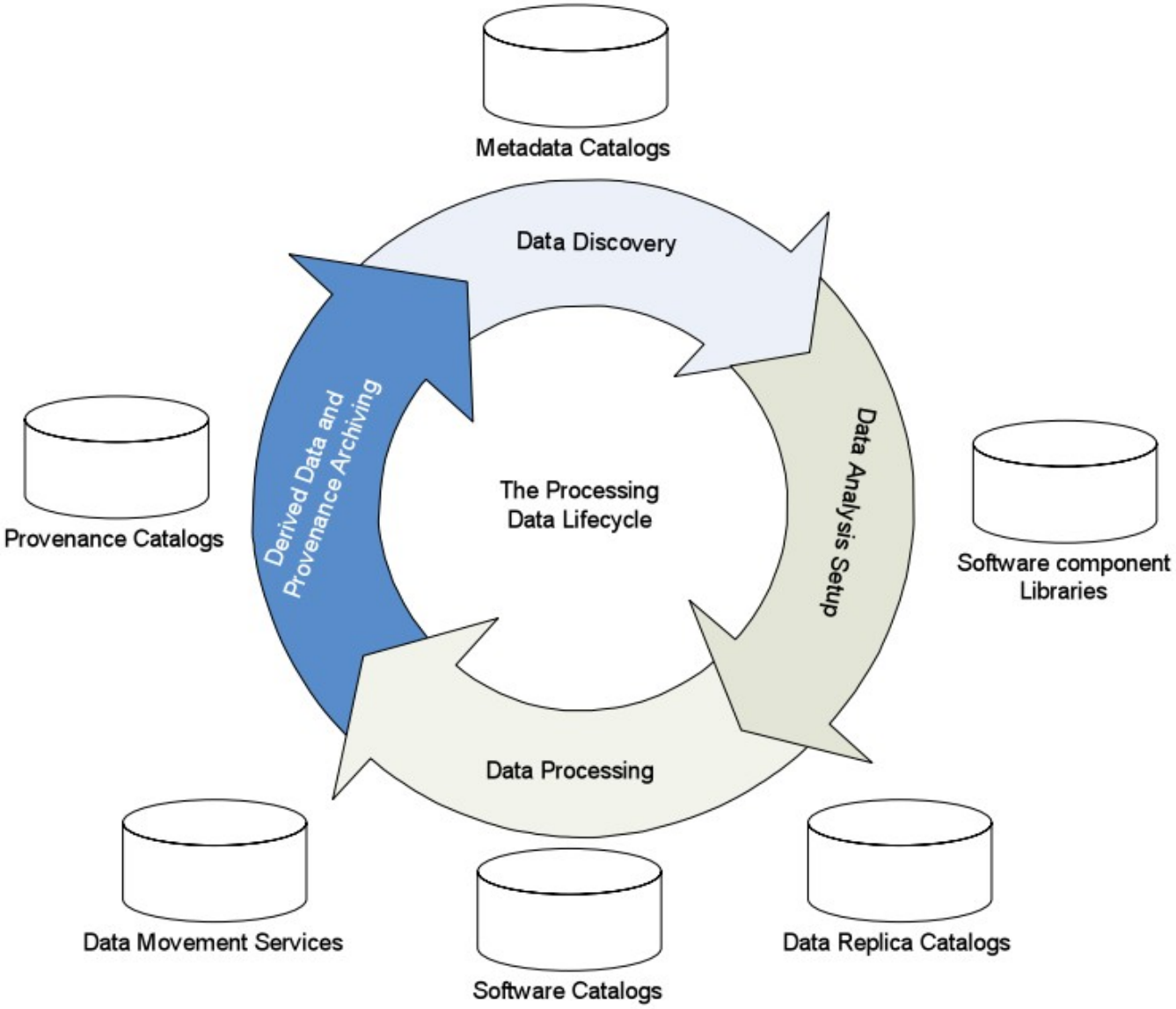
TensorFlow



Metadata and provenance

- Today, the key drivers for the capture and management of data descriptions are the **scientific collaborations**
  - They bring collective knowledge and resources to explore a research area
- These data need to contain enough information so that members of the collaboration can interpret them and use them for their research
- Metadata and provenance information are also important for the automation of scientific analysis
  - Analysis software needs to
    - be able to identify the datasets appropriate for a particular analysis
    - Annotate new, derived data with metadata and provenance information

The data lifecycle



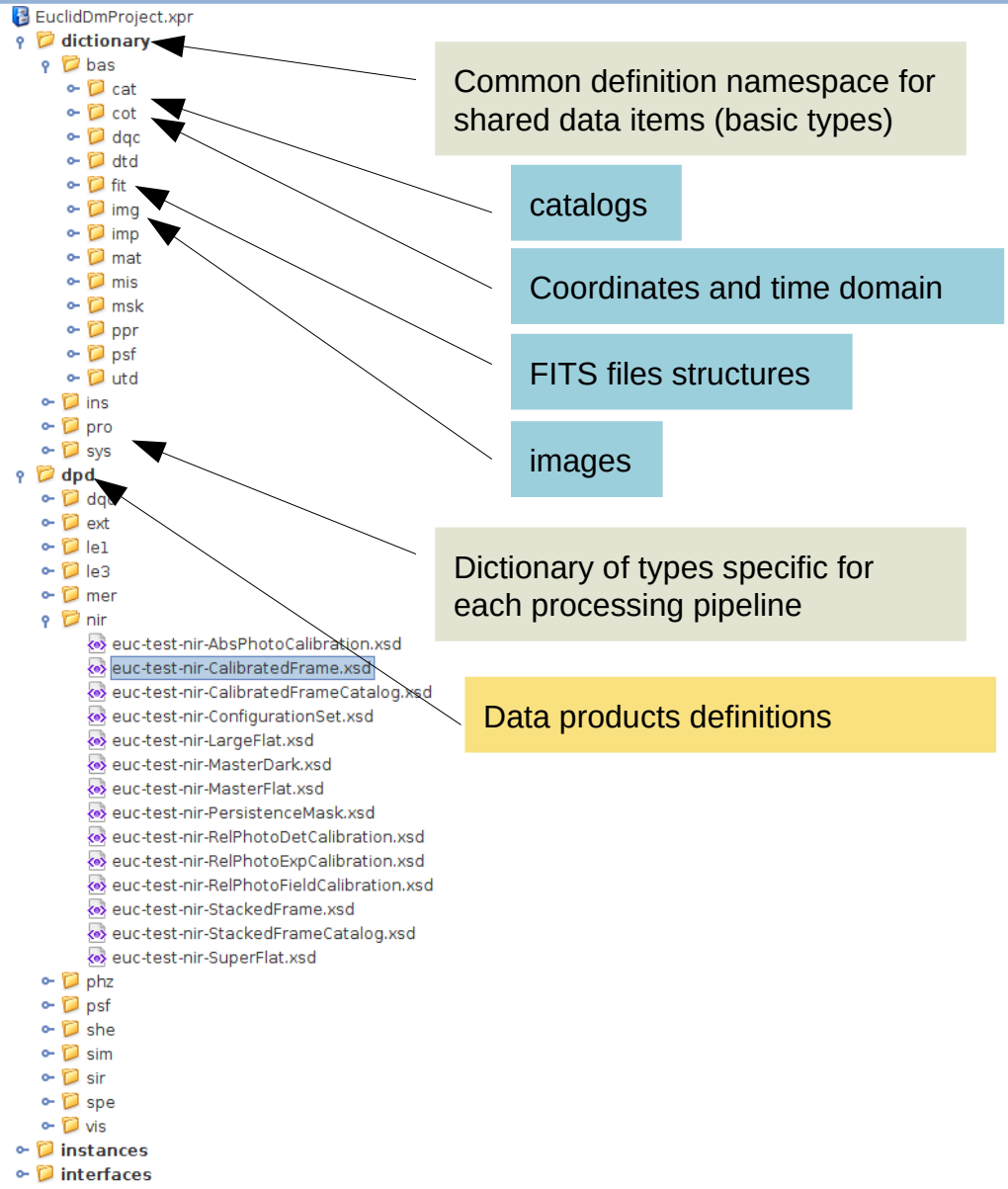
Metadata definition

- **Metadata**: “structured data about an object that supports functions associated with the designated object”
- *Structured data* implies a systematic ordering of data according to a metadata schema specification (see "**Metadata Vocabulary**")
- *Functions associated with the designated object*. The emphasis here is on the ability of metadata to support the activities and behaviors of an object. For example:
  - For example, ‘‘author,’’ ‘‘title,’’ and ‘‘subject’’ metadata facilitate the *discovery* of an information resource
  - An ‘‘invoice number’’, ‘‘product code’’, ‘‘credit card number (for payment)’’ and ‘‘date of financial transaction” metadata capture the *purchase* activity for a consumer good
- The above definition covers the dual function of the metadata:
  - describing the objects from a logical point of view
  - describing their physical and operational attributes.

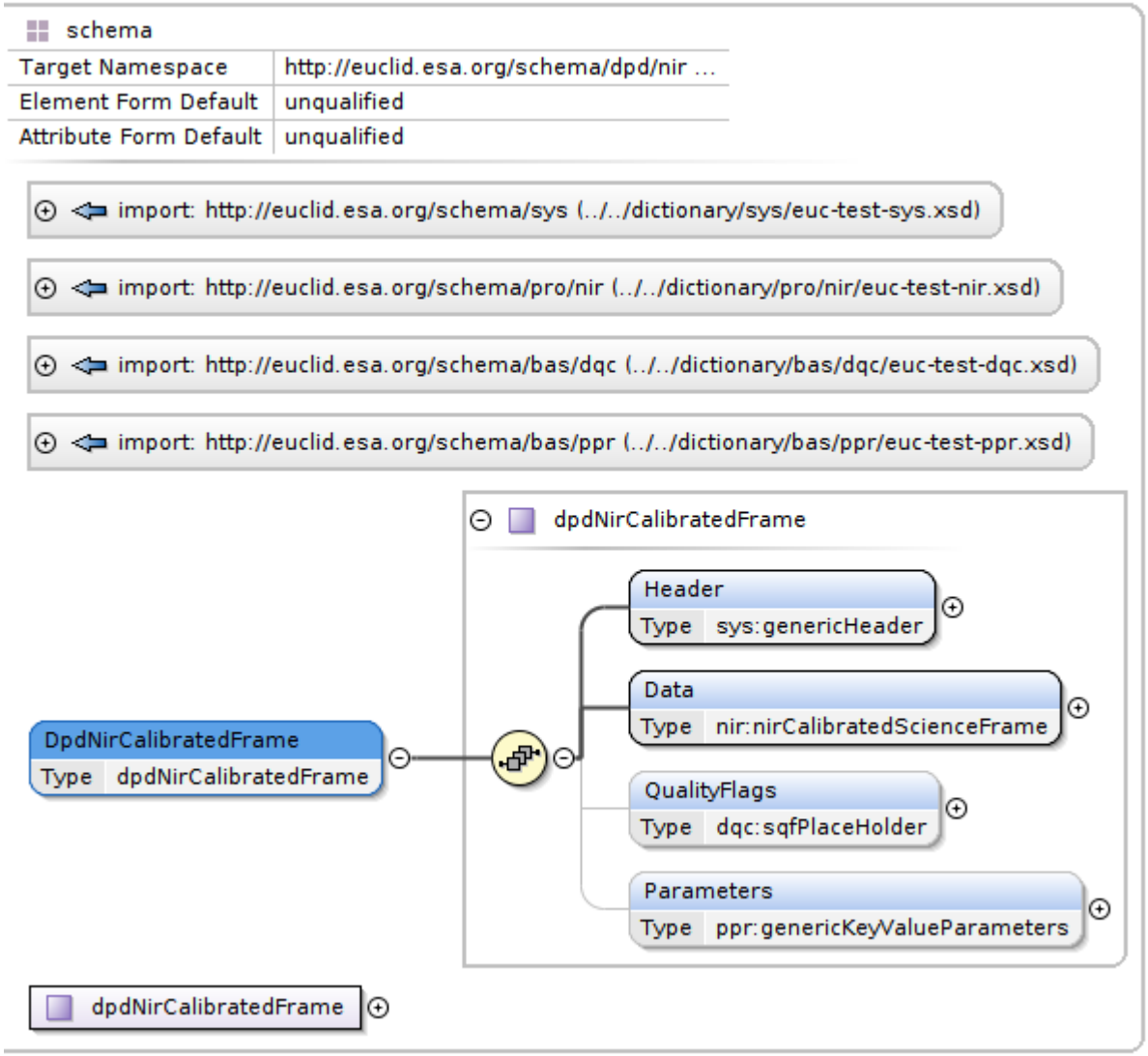
Metadata layers

- Metadata can be organized in layers. It can refer to:
  - raw data, e.g. coming from an instrument
  - information about the process of obtaining the raw data
  - derived data products
- This allows distinguishing different layers (or chains) of metadata: primary, secondary, tertiary, and so forth.
- Example with the satellite imaging domain: raw images taken by instruments in the satellite are sent to the ground stations.
  - Primary metadata includes:
    - the times when images were obtained and transferred
    - the instrument used to acquire them
    - The position to which each image refers
  - Secondary metadata:
    - Checking for gaps in the acquired data
    - Grouping of primary metadata information for a given instrument
    - Quality of the data in a given time period
    - Statistical summaries

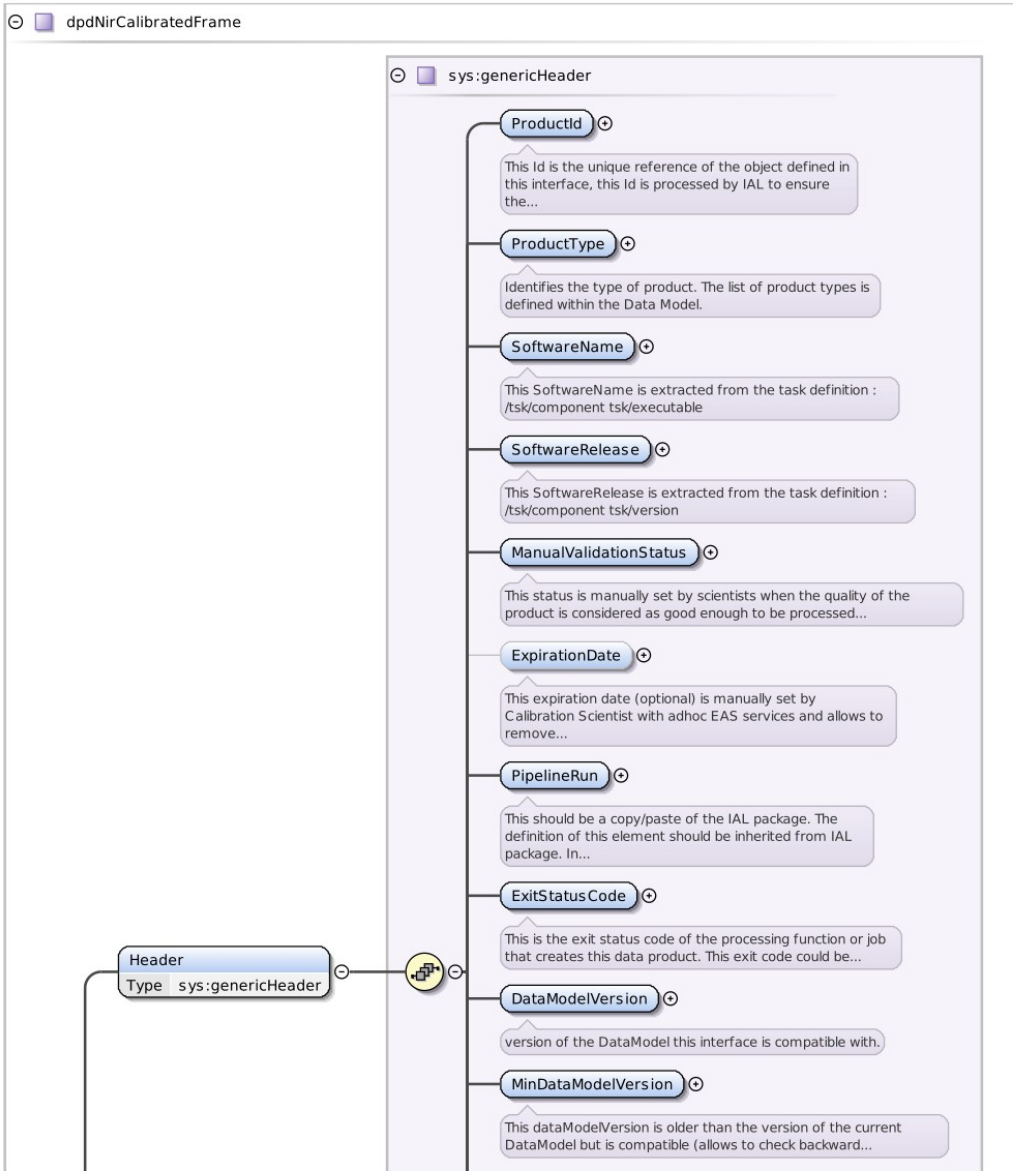
Example: Euclid project data model



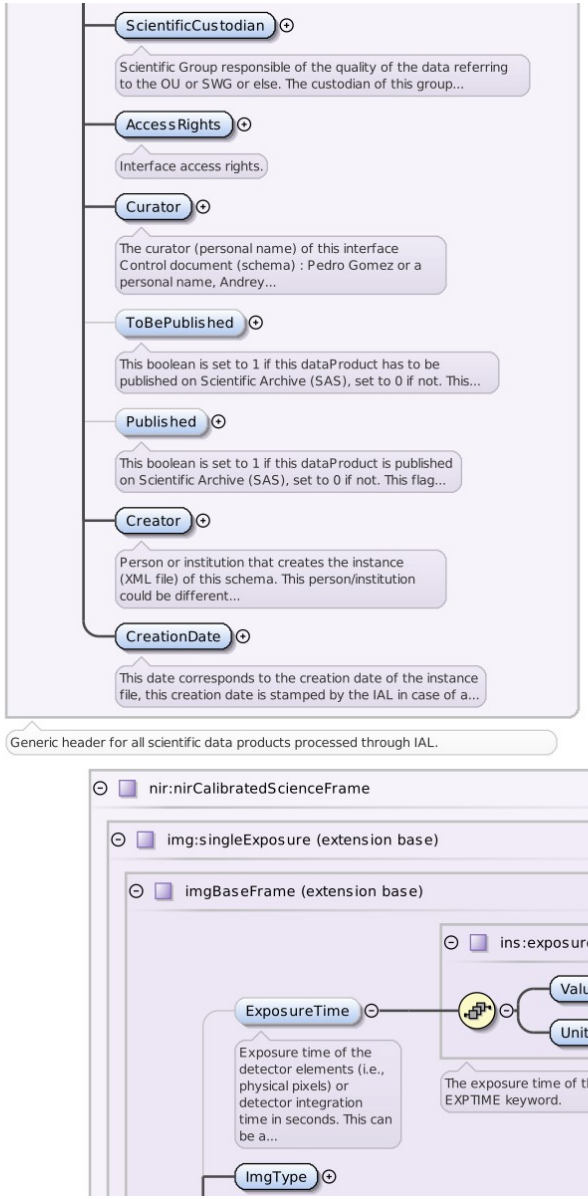
Example: Euclid project data model



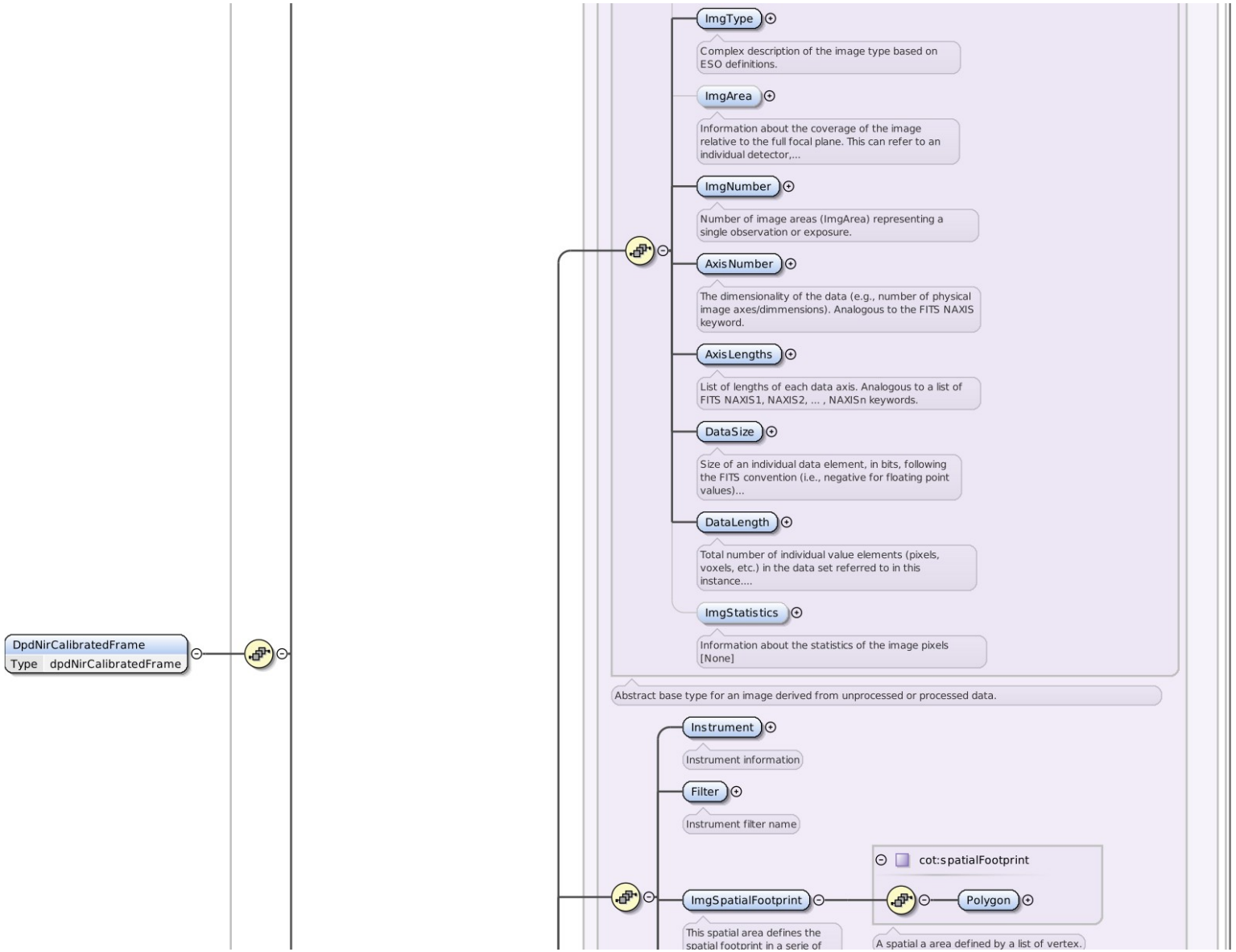
Example: Euclid project data model



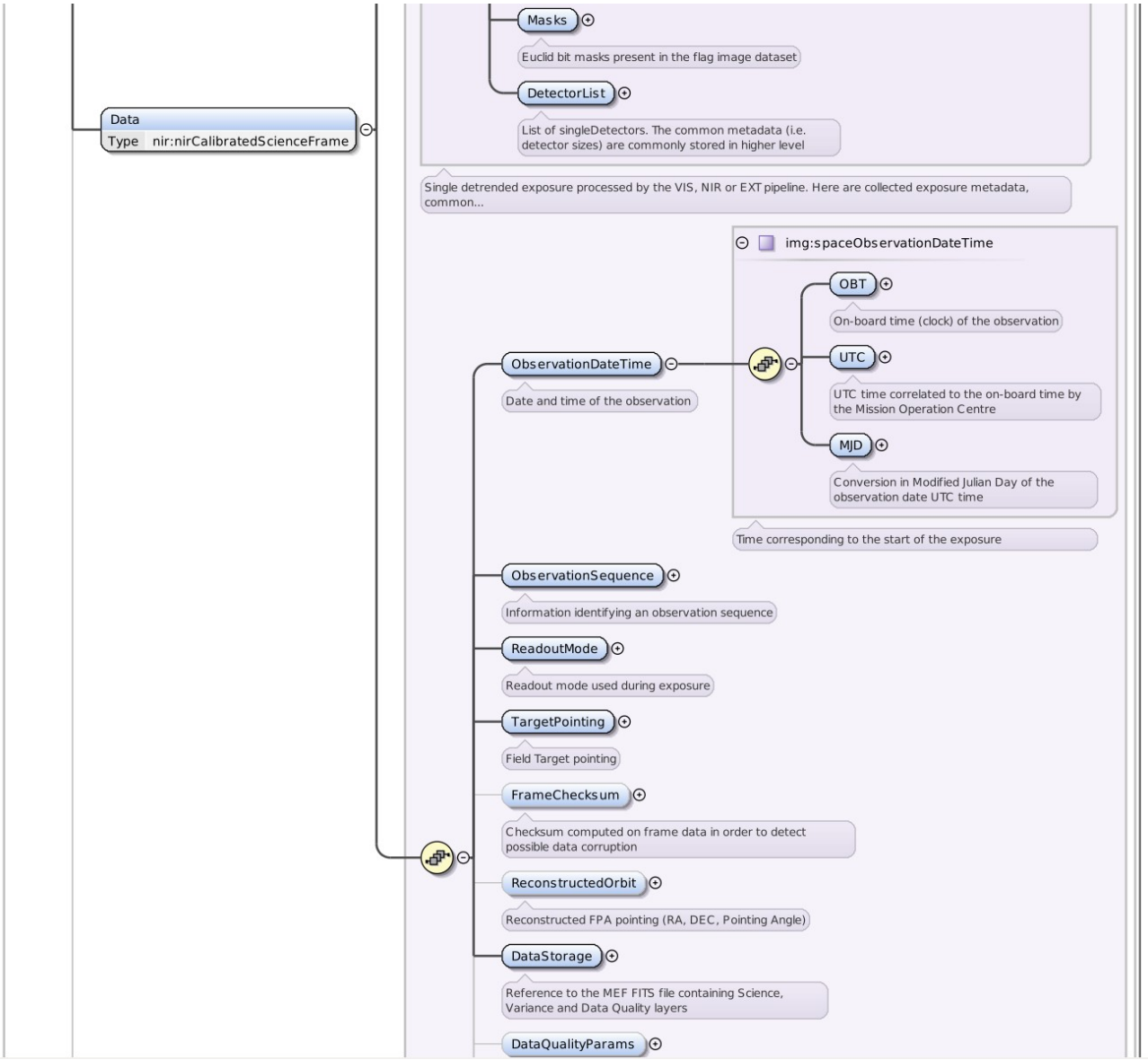
Example: Euclid project data model



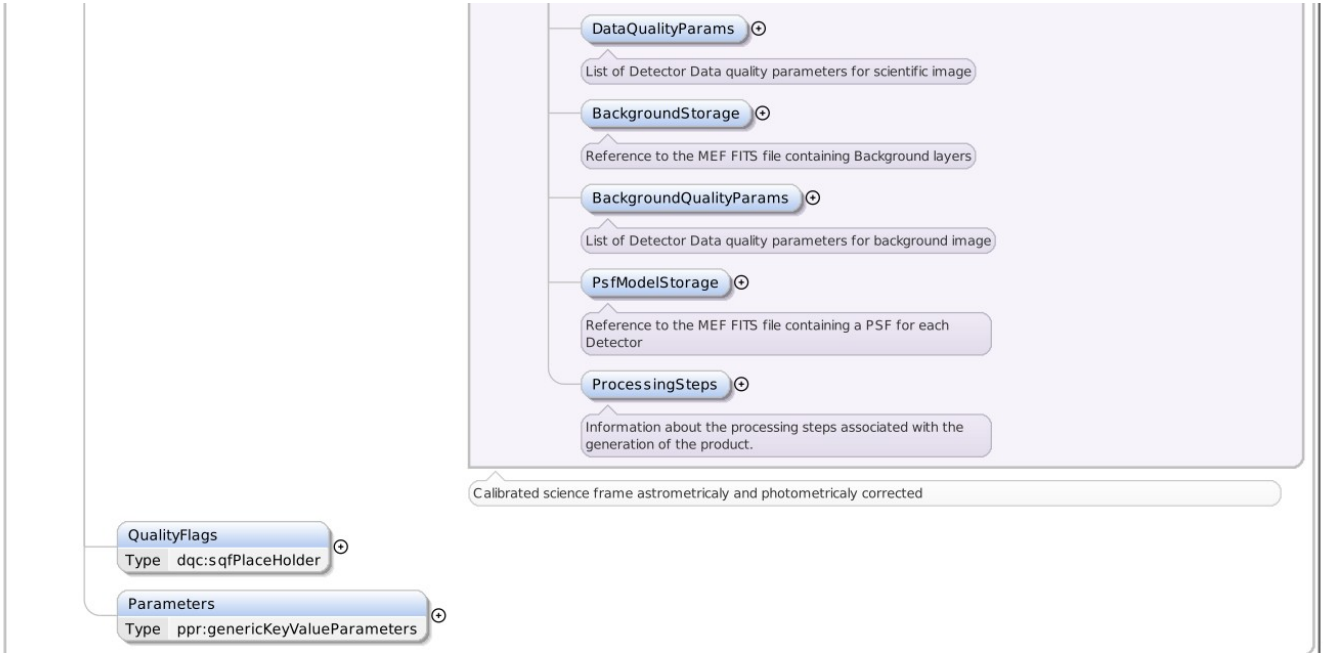
Example: Euclid project data model



Example: Euclid project data model



Example: Euclid project data model



Ontologies

- Together with controlled vocabularies and thesauri, **ontologies** have become one of the most common means to specify the structure of metadata in scientific applications
- Ontologies are normally defined as “formal, explicit specifications of shared conceptualizations”
- A *conceptualization* is an abstract model of some phenomenon in the world derived by having identified the relevant concepts of that phenomenon.
- *Explicit* means that the type of concepts used, and the constraints on their use, are explicitly defined
- *Formal* refers to the fact that the ontology should be machine readable
- *Shared* reflects the notion that an ontology captures consensual knowledge; that is, it is not private view for some individual, but accepted by a group

Ontologies formalization

- Not all ontologies have the same degree of formality;
- Given this fact, ontologies are usually classified either as lightweight or heavyweight.
- An example of the former would be *Dublin Core*, which is being widely used to specify simple characteristics of electronic resources
  - it specifies a predefined set of features such as creator, date, contributor, description, format, and the like
- An example of the latter is the Ontology of Astronomical Object Types (see next slides)
- Lightweight ontologies can be specified in simpler formal ontology languages like the **Resource Description Framework** (RDF) Schema
- Heavyweight ontologies require more complex languages like the **Web Ontology Language** (OWL)

Dublin Core example

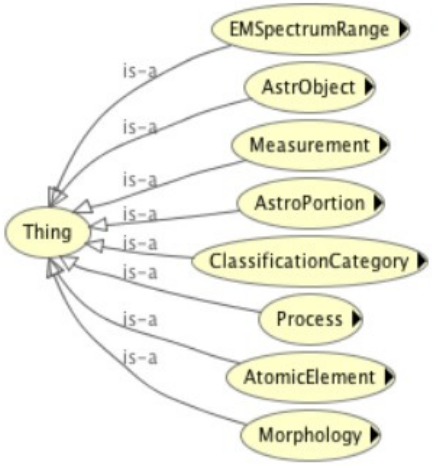
- Dublin Core definition with the RDF Schema

```
<rdf:Description rdf:about="http://purl.org/dc/elements/1.1/title">
    <rdfs:label xml:lang="en">Title</rdfs:label>
    <rdfs:comment xml:lang="en">A name given to the resource.</rdfs:comment>
    <rdfs:isDefinedBy rdf:resource="http://purl.org/dc/elements/1.1/"/>
    <dcterms:issued rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
       1999-07-02
    </dcterms:issued>
    <dcterms:modified rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
       2008-01-14
    </dcterms:modified>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <dcterms:hasVersion rdf:resource="http://dublincore.org/usage/terms/history/#title-006"/>
</rdf:Description>
```

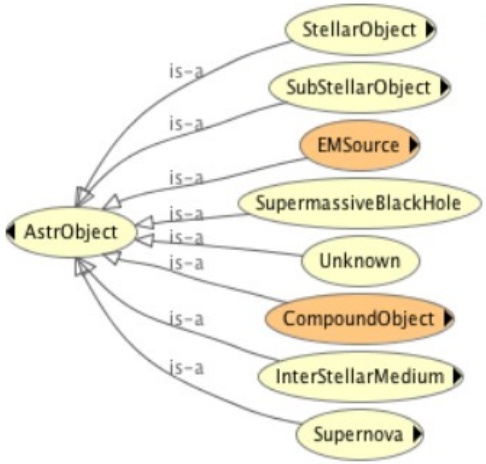
- Mainstream developers have used, and continue to use, vocabularies such as Dublin Core in the context of relational databases and repositories, many of which are based on XML

Ontology of Astronomical Objects Types

4.2.1Top-level concepts



The AstrObject section



http://www.ivoa.net/documents/Notes/AstrObjectOntology/

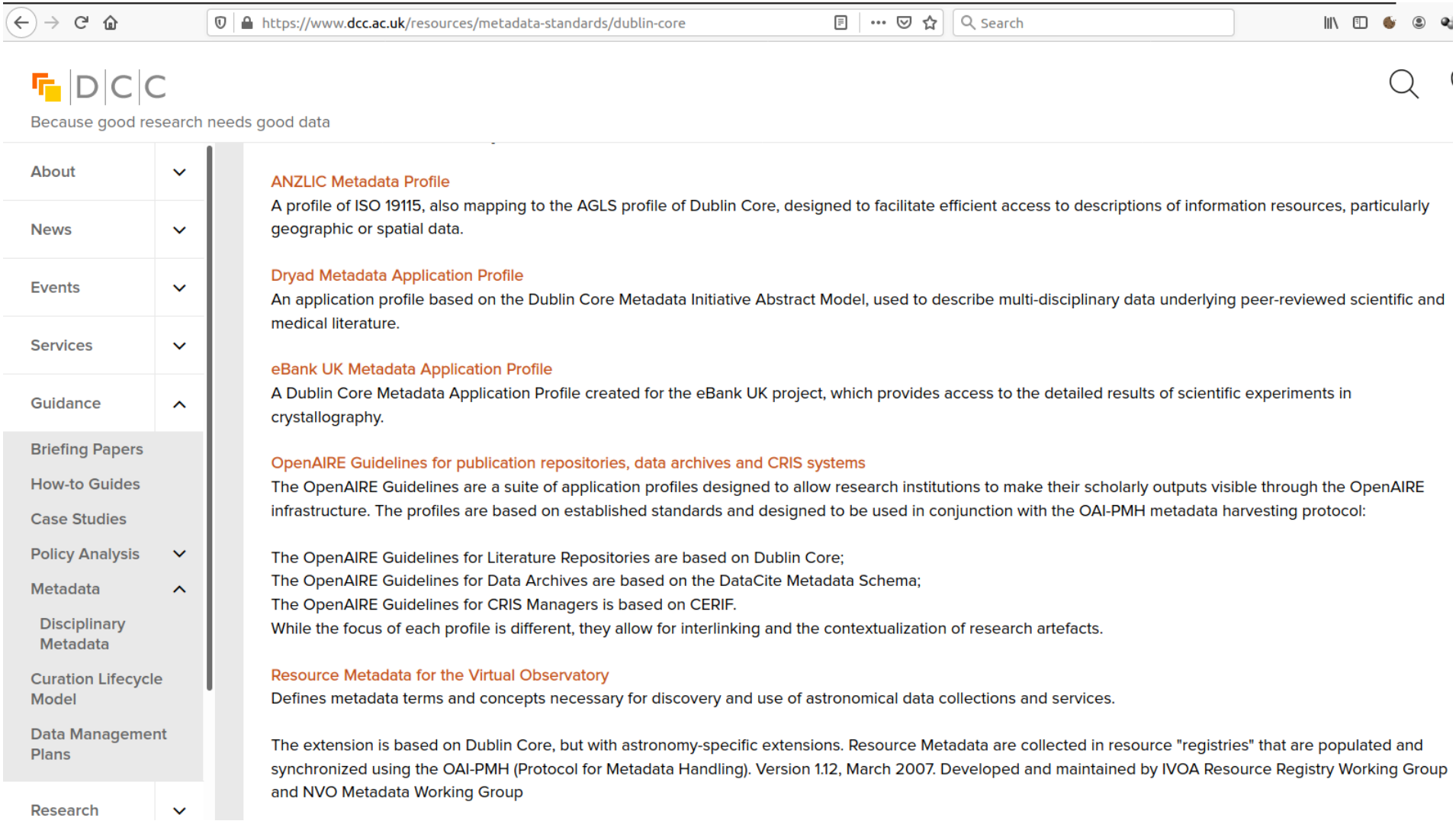
OWL specification:
http://wiki.ivoa.net/internal/IVOA/IvoaSemantics/ObjectTypes_0.99r.owl

DoubleStar ≡ AstrObject and hasComponent exactly 2
and hasComponent only StellarObject

Metadata schema and attributes

- Metadata attributes that are elements of a metadata schema can encompass a variety of information
- Some metadata is **application independent**, such as the creation time, and author, as described in Dublin Core
- Other metadata is **application dependent** and may include attributes such as the duration of an experiment, temperature of the device, and others
- Many applications have expanded the Dublin Core schema to include application-dependent attributes (see next slide)
- Definitions provided in the Dublin Core or its extensions may be instantiated in a variety of standard forms, e.g. XML, and with a variety of mechanisms, e.g. OWL, or RDBMSs
  - Consequently, the **exact names** and rendering of the **values** may depend on the particular form in which they are represented

Dublin core extensions



https://www.dcc.ac.uk/resources/metadata-standards/dublin-core

Search

DCC

Because good research needs good data

About
News
Events
Services
Guidance
Briefing Papers
How-to Guides
Case Studies
Policy Analysis
Metadata
Disciplinary Metadata
Curation Lifecycle Model
Data Management Plans
Research

ANZLIC Metadata Profile
A profile of ISO 19115, also mapping to the AGLS profile of Dublin Core, designed to facilitate efficient access to descriptions of information resources, particularly geographic or spatial data.

Dryad Metadata Application Profile
An application profile based on the Dublin Core Metadata Initiative Abstract Model, used to describe multi-disciplinary data underlying peer-reviewed scientific and medical literature.

eBank UK Metadata Application Profile
A Dublin Core Metadata Application Profile created for the eBank UK project, which provides access to the detailed results of scientific experiments in crystallography.

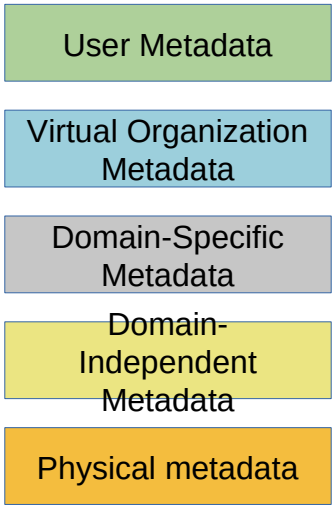
OpenAIRE Guidelines for publication repositories, data archives and CRIS systems
The OpenAIRE Guidelines are a suite of application profiles designed to allow research institutions to make their scholarly outputs visible through the OpenAIRE infrastructure. The profiles are based on established standards and designed to be used in conjunction with the OAI-PMH metadata harvesting protocol:

The OpenAIRE Guidelines for Literature Repositories are based on Dublin Core;
The OpenAIRE Guidelines for Data Archives are based on the DataCite Metadata Schema;
The OpenAIRE Guidelines for CRIS Managers is based on CERIF.
While the focus of each profile is different, they allow for interlinking and the contextualization of research artefacts.

Resource Metadata for the Virtual Observatory
Defines metadata terms and concepts necessary for discovery and use of astronomical data collections and services.

The extension is based on Dublin Core, but with astronomy-specific extensions. Resource Metadata are collected in resource "registries" that are populated and synchronized using the OAI-PMH (Protocol for Metadata Handling). Version 1.12, March 2007. Developed and maintained by IVOA Resource Registry Working Group and NVO Metadata Working Group

Metadata types



- At the lowest level , **physical metadata** includes information about the physical storage systems as well as replica location metadata
- **Domain-Independent metadata** includes generic attributes, such as logical names, creator, modifier, data content, authorization, etc.
- **Domain-Specific metadata** attributes are often defined by metadata ontologies developed by the application communities
- A **virtual organization** that includes multiple scientific institutions may define attributes for characterizing data sets
- **Individual users** may want to associate metadata attributes such as annotations to data items or collections

Domain-Independent metadata

- We can identify a number of logical categories for domain-independent metadata
- In the following we provide categories related to data file handling:
  - Logical File metadata
  - Logical collection metadata
  - Logical view metadata
  - Authorization metadata
  - Audit metadata
  - Creation and transformation history metadata (provenance)

Logical File Metadata

- A **logical file name** attribute specifies a name that is unique within the namespace managed by a metadata service
- A **data type attribute** describes the data item type, for example, whether the file format is binary, html, XML, and so forth
- A **valid attribute** indicates whether a data item is currently valid
- If data files are updated over time, a **version attribute** allows us to distinguish among versions of a logical file
- A **collection identifier** attribute allows us to associate a logical file with **exactly one** logical collection
- **Creator** and **last modifier attributes** record the identifications of the logical file's creator and last modifier

Logical collection and Logical view

- **Logical collection** metadata attributes and purpose
  - A **collection name** and a **description** of the collection, which consists of a collection of other logical files and other logical collection
  - The **creators** and **modifiers** of the collection
  - **Audit** information
  - Optionally, a **parent attribute** that records the identifier of the parent logical collection
  - One of the purpose of the logical collections is to **support authorization on groups** of files rather than on individual files
- **Logical view** metadata:
  - **Logical view name** and **description**
  - Information about logical files, logical collections and other logical views that compose this logical view
  - Logical views **do not affect authorization**, which is enforced with rights defined for logical collections and individual logical files

Authorization metadata

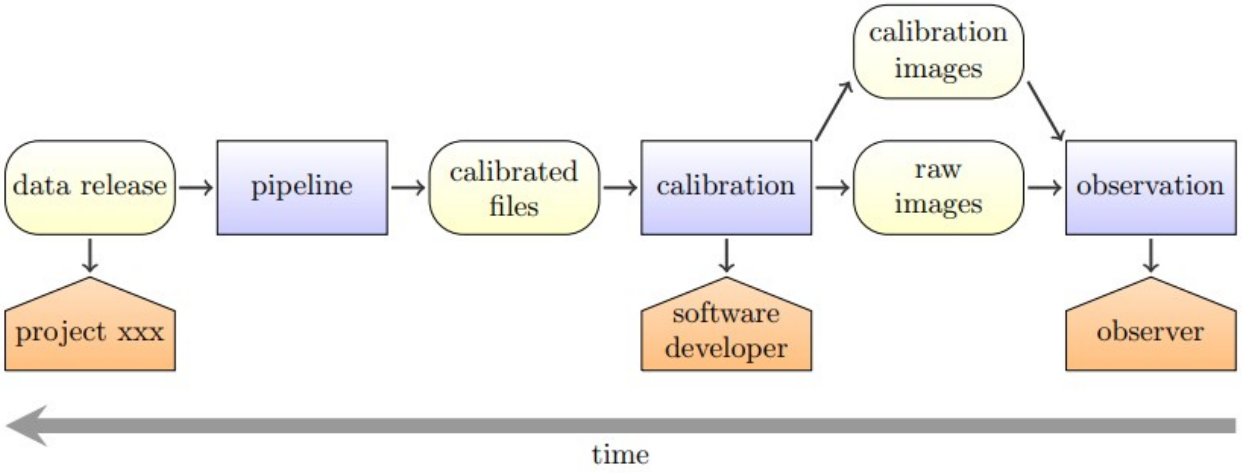
- **Authorization** metadata attributes specify access privileges on logical files, collections and views
- If an **external authorization service** is used, then authorization information must also be maintained for the external service
- **Access permissions** specified on a logical collection apply to all logical files that comprise that logical collection (and its sub-collections)
- The effective set of permissions on a logical file is the union of the permissions on that file and the permissions on a logical collection to which the file belongs, and so on up the hierarchy of collections

User and Audit metadata

- **User metadata** provide attributes that describe writers of metadata, including contact information.
- The attributes specify the distinguished name, description, institution, address, phone, and email information for writers.
- **Audit metadata** is used to record actions performed via the metadata service
  - An **audit record** includes the attributes that specify the object identifier upon which an action was performed and the object type (logical file, logical collection or logical view)
  - Audit metadata also includes a text description of the audited action as well as the distinguished name (DN) of the user who performed the audited action
  - Finally, the audit record contains the timestamp at which the audited operation was performed.

Provenance 1/2

- The scope of the provenance is mainly modeling of the flow of data, of the relations between data, and of processing steps



An example graph of provenance discovery. Starting with a released dataset (left), the involved activities (blue boxes), progenitor entities (yellow rounded boxes) and responsible agents (orange pentagons) are discovered.

Provenance 2/2

- Provenance information may be recorded in minute detail or by using coarser elements, depending on the intended usage
- The provenance metadata can
  - Provide information on which steps were taken to produce a dataset and list the methods/tools/software that were involved.
  - Provide information on the experimental conditions
  - Track the history back to the raw data files / raw images, show the workflow (backwards search), or return a list of progenitor datasets
  - Identify the people or organizations involved in the production of a dataset
- The provenance metadata can help:
  - Find the location of possible error sources in the generation of a dataset
  - Judge the quality of an observation, production step or dataset
  - Locate derived datasets or outputs (forward search)

Approaches to data integration

- More and more domain-specific data management infrastructures are built to allow users **easy access to scientific data**, often through comprehensive web portals
- Traditional **data integration** basically follows a schema-matching approach in which related schema components (relations and attributes) from different sources are identified and homogenized
- Integration aims at providing a single conceptual view over the data managed at sources
- Using this view, the data can either be physically integrated at a single site (physical integration of data)
- Or the data can be queried in a uniform and transparent fashion. This approach results in a **federated** or multi-database system (logical integration)

Integration through Interoperability

- Interoperability among heterogeneous and **distributed data sources** is a fundamental requirement not only in the context of scientific data management, but in any type of distributed computing infrastructure.
- **Interoperability** is generally defined as “the ability of two or more systems or components to exchange information and to use the information that has been exchanged.”
- Interoperability means that systems can **exchange information** and data using **standard protocols** and **formats**
- Interoperability among data repositories and applications has become a main driver to facilitate scientific data management and exploration on a large scale
- Grid computing infrastructures have significantly contributed to this development
- A more recent trend in these science initiatives is to increase interoperability aspects through **service oriented science**

Service oriented architectures

- **Service-oriented architecture** (SOA) allows an effective cooperation among data sources and data processing components hosted at different organizational units
- SOA supports reusability and interoperability of software and service components on the Web, thus increasing the efficiency of developing and composing new services
- In a SOA-based system, all data and process components are modeled as **Web services**
- Web services can be implemented using different technologies:
  - Web Services Description Language (WSDL) and Simple Objects Access Protocol (SOAP)
  - As RESTFul web services
  - XML-RPC

Registry (and catalog) services

- As scientific data are accumulating at an ever-increasing speed, it is very difficult if not impossible for users to know exactly the details of all the data that might be relevant to their project
- A **registry service** enables users—human or software—to locate, access, and make use of resources in an open, distributed system
  - it facilitates the retrieval, storage, and management of many kinds of resource descriptions
- Based on SOA, a registry service must support some fundamental interactions:
  - **publishing** resource descriptions so that they are accessible to prospective users
  - **discovering** resources of interest according to some set of search criteria
  - and then **interacting** with the **resource provider** to access the desired resources
- The terms ‘catalogue’ and ‘registry’ are often used interchangeably
- A registry is a specialized catalogue that exemplifies a formal registration process
- A registry is typically maintained by a registration authority, who assumes responsibility for complying with a set of policies and procedures for accessing and managing registry content

Example: geospatial data

- The use of geospatial data obtained through observations and simulations and their management in spatial data infrastructures have become essential in many application domains.
  - environmental monitoring, climate research, disaster prevention, natural resource management, transportation, etc.
- The types of geospatial data considered in these domains come in a variety of types
  - imagery from air and space-borne instruments
  - vector data describing geographic objects and features
  - outputs from simulations
  - and numerous types of real-time sensor data
- The two most common approaches to model geographic information are using either an object-based model or a field-based model.

Object-based and field-based models

- In an **object-based** model, geographic objects correspond to real-world entities (also called features)
- A feature typically has two parts:
  - a spatial component (or spatial extent), which specifies the shape and location of the object in the embedding space;
  - a descriptive component that describes the non-spatial properties of the feature in the form of attributes
- The spatial extent of an object is typically modeled as a point, polyline, or polygon
- In **field-based** approaches, the space to be modeled is partitioned (tessellated) into two- or multidimensional cells, a cell having a spatial extent.
  - With each cell one or more attribute values are associated, each attribute describing a continuous function in space
  - E.g.: multi-spectral raster imagery from remote sensors

Geospatial schemas

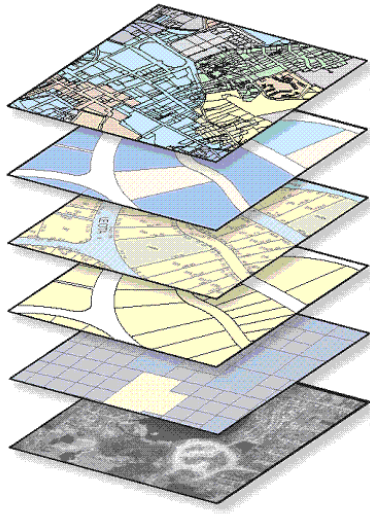
- Essential ingredients to any data integration approach: information about the schemas as well as metadata for schema components and the data managed in heterogeneous scientific data repositories
- The Geography Markup Language (GML) is an XML-based specification for representing geographic features
- GML serves as an open interchange format for geospatial data as well as a modeling language for geographic information
  - the later versions are based on XML-Schema
  - GML application schemas are very flexible in that they allow users to tailor and extend predefined GML data types
  - It also serves as data exchange format for geospatial data

Geospatial metadata

- There are many metadata frameworks for spatial data and applications that have geospatial components
- The Content Standard for Digital Geospatial Metadata (CSDGM) has entries to describe
  - the geographical area a geospatial dataset covers
  - the spatial data model that is used to encode the spatial data (vector/raster) or other possible methods for indirect georeferencing
  - the information about the spatial reference system
- In addition to the CSDGM, several other metadata standards have been developed over the past few years for different application domains in the geosciences and environmental sciences

Integrating geospatial data: overlay

- The most common view of this is to have a Geographic Information Systems (GIS) that allows users to overlay different themes (layers)
  - several georeferenced themes that represent different characteristics of that area
  - A theme can be represented by either vector data or field-based data

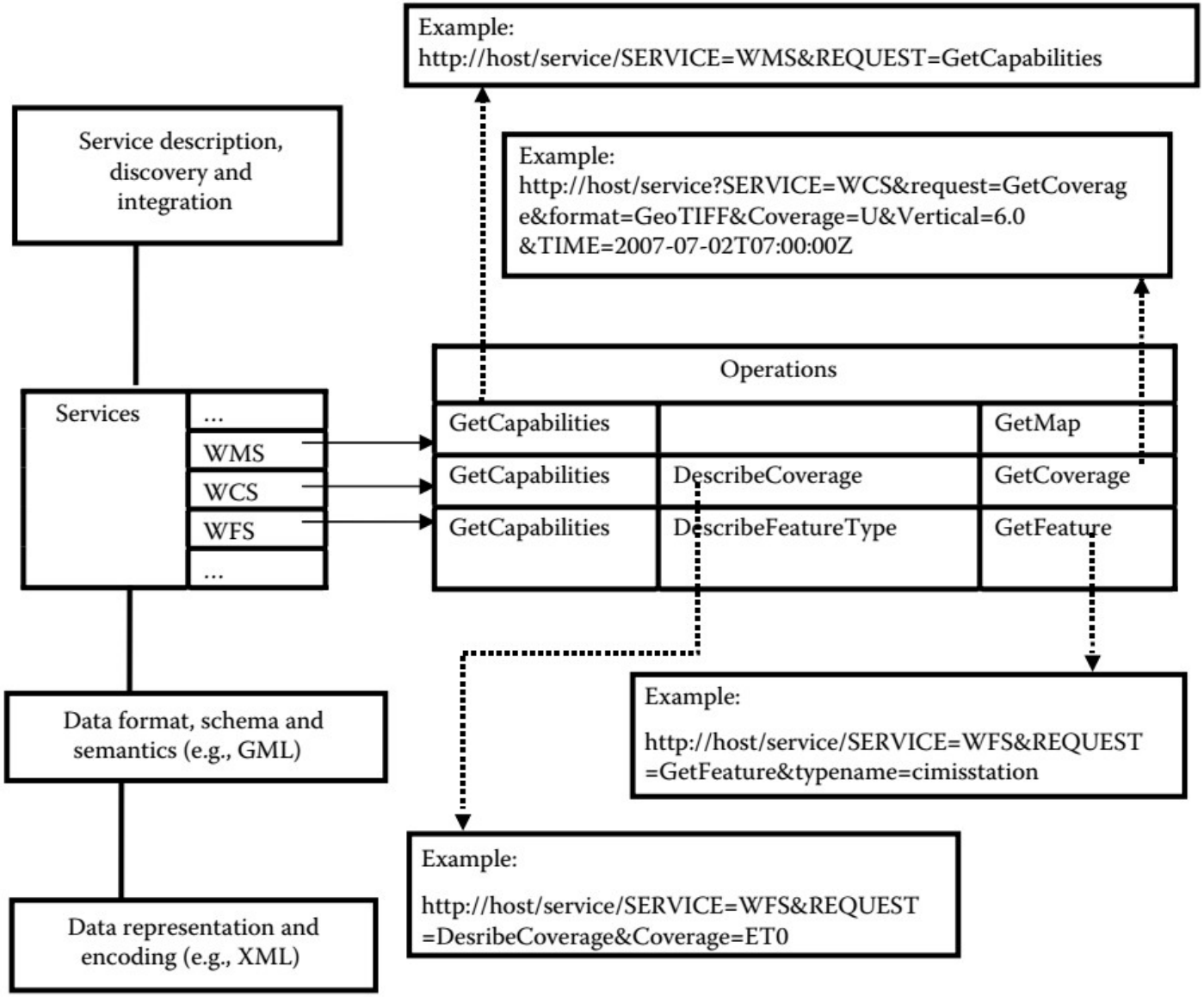


- Re-projection, georeferencing, and scaling are tasks that are typically performed on the datasets prior to their overlay or integration
- The key in dealing with conflicting spatial components is to make use of the location information associated with geospatial objects
- it is important to have a spatial reference system (SRS) (or coordinate reference system (CRS)) underlying the space in which features and phenomena are modeled
- The Open Geospatial Consortium (OGC) was founded with the mission of advancing the “development of international standards for geospatial interoperability

Catalog Service for the Web

- In the context of geospatial applications, OGC's Catalog Service for the Web (CSW) Implementation Standard provides several operations that allow users to interactively or programmatically search and retrieve metadata that are related to the use of the datasets
- The mandatory **GetCapabilities** operation returns metadata about the specific repository server (ServiceIdentification)
  - the operations supported by the service including the URL(s) for operation requests (OperationMetadata)
  - The type of resource cataloged by the repository server (Content)
  - and the query language and its functionality supported by the repository
- The **GetRecords** operation allows users to specify query constraints and metadata to be retrieved and returns the number of items in the result set and/or selected metadata for the result set
- The **DescribeRecord** operation allows a client to discover elements of the information model supported by the target catalog service
- The optional **GetDomain** operation is used to obtain run-time information about the range of values of a metadata record element
- The mandatory **GetRecordByID** request retrieves the default representation of catalog records using their identifier
- With the above operations, users are able to probe the repository server's capabilities, search the repository, negotiate the format of the metadata and finally retrieve the metadata of the dataset(s) of interest

Example of OGC standard services



OCG standard services

- Web Feature Service (WFS): WFS defines interfaces for querying and retrieving features based on spatial and nonspatial properties of the features (object-based data)
  - Data between the service and the client is exchanged in GML
- Web Coverage Service (WCS): WCS defines interface to query and retrieve spatially referenced coverages (gridded or raster data)
- Web Map Service (WMS): This service produces maps (in the form of digital images) of spatially referenced data (i.e. features or coverages) from a data source managing geographic information
- The realization of the above services typically occurs in the form of **middleware layers** that clients can access through the Web
  - Representatives of such middleware layers are the open source systems GeoServer 71 and MapServer