A Metadata Catalogue – not only for Lattice Data

Basavaraja BS, Hubert Simma

TA2 Meeting

30 September 2022

Overview

1. Rationale
2. Implementation
3. Status and Plans

Rationale behind a MDC

FAIR principles, in particular — Wilkinson 2016

F1	globally unique and persistent ID assigned to (M)D	
F4	(M)D registered or indexed in a searchable resource	→ MDC
A1	(M)D retrievable by ID using standardized protocols	
A1.2	protocol allows authentication/authorization procedure where necessary	
A2	MD accessible even if data is no longer available	

- conceptually refer to three types of entities:
  - data = any digital object
  - metadata (MD) = information about a digital object
  - infrastructure
- define guiding principles, not an implementation

MDC Implementation

Assumption: data objects may be large (while volume of MD is moderate)

thus

- data storage is usually distributed
- efficient searching requires metadata to be stored separately from data
- resource for registration of (meta)data can be centralized ($\rightarrow$ MDC)

Logically the MDC is a database with

- ID = primary key
- MD schema = well-defined (and flexible) structure of attribute values
  (with possibly different schemas for different collections of (meta)data)

and implemented as a webservice with API contract + client tools for

- search (F4): Query $\rightarrow$ Set of ID
- retrieval (A1): ID $\rightarrow$ MD (in different representations/formats)

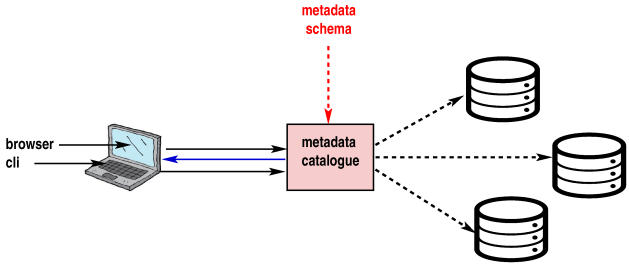
MDC Use-case in ILDG

Metadata

- follows a well-defined and rich schema
- stored separately from (big) data
- searchable in central catalogue(s)

MDC supports

- validation of MD against a (freely configurable) MD schema
- flexible search in MD content
- relations between different data entities
- fine-grained access control for metadata (and data)



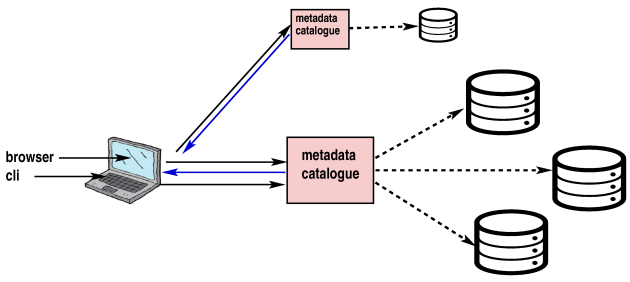
MDC Use-case in ILDG

Metadata

- follows a well-defined and rich schema
- stored separately from (big) data
- searchable in central catalogue(s)

MDC supports

- validation of MD against a (freely configurable) MD schema
- flexible search in MD content
- relations between different data entities
- fine-grained access control for metadata (and data)



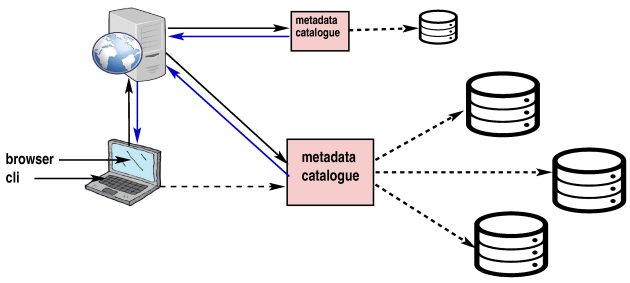
MDC Use-case in ILDG

Metadata

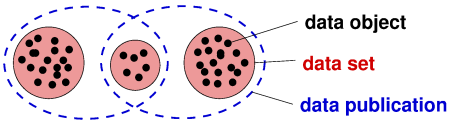
- follows a well-defined and rich schema
- stored separately from (big) data
- searchable in central catalogue(s)

MDC supports

- validation of MD against a (freely configurable) MD schema
- flexible search in MD content
- relations between different data entities
- fine-grained access control for metadata (and data)



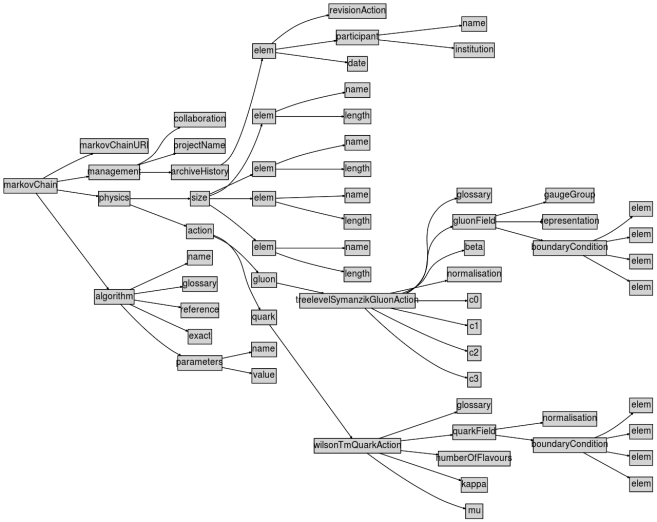
Identifiers and MD Schemas in ILDG



entity	ID	MD includes information on			
		relations	content	data storage	access control
data objects	`lfn`	unique mcu	✔	✔	—
↓					⇑
data sets	`mcu`	—	✔	—	✔
↖ ↑ ↗					
data publication	*doi*	set of mcu	✔	—	—

Tree structure of content metadata in ILDG

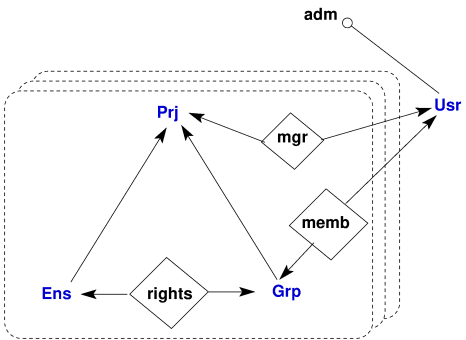
QCDmlEnsemble.xsd



Fine-grained access control model

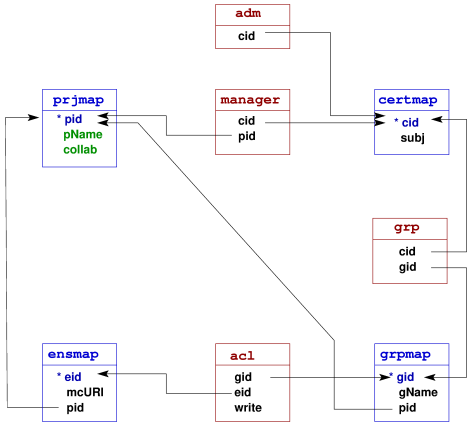
- Each data set (Ens) is associated with a unique project (Prj)
- Each project has a corresponding group of users with manager rights (mgr)
- Managers can define additional user groups (Grp) for the corresponding project
- Each user group has specific access rights (R/W) for the (meta)data sets of the project

Entity relationships:



Attribute service of MDC

Tables (with API for CRUD operations):



N.B.: pName and collab *must* match corresponding entries in `managment` part of metadata

Status and Plans

MDC of ILDG is running since 2008, but

- uses SOAP API and out-dated software
- supports only 2 MD schemas
- AA is based on grid certificates and VO attributes

web interface

WSDL

ILDG

VOMS

Step 1: Re-build (in progress), e.g. for (test) instances in PUNCH

- container-ready (Docker image and Helm scripts)
- deployment in Kubernetes (or other) cluster

Step 2: Re-design (or adaption of other designs)

- REST API
- multiple MD schemas
- token-based AA