ILDG Demo

Lattice 2023

ILDG Working Groups

August 4, 2023

Overview

1. Links
2. Search and Download Metadata
3. Download Configurations
4. Upload Configurations

Links

- ILDG Web Page, see https://hpc.desy.de/ildg/
  - → Specifications (XML schemas, file format)
  - → Services (URLs)
  - → Organization (Working Groups, Meetings)
- Material from Hands-on Workshop June 2023
  - instructions and lectures https://indico.desy.de/event/39311/
  - material from exercises https://gitlab.desy.de/ildg/hands-on/material.git
  - simple client tools https://gitlab.desy.de/ildg/hands-on/try-client.git
  - container environment https://gitlab.desy.de/ildg/hands-on/workshop-image.git
- Metadata
  - MDWG web page → `document` https://www2.ccs.tsukuba.ac.jp/ILDG/
  - Markup tool https://www.jldg.org/QCDml/
- Simple MDC and FC listings, e.g. https://hpc.desy.de/ldg/metadata_indices/
- Contact email (Working Groups, etc.) ildg-contact(at)desy.de

Search and Download Metadata

Prerequisites:

- ❒ Web browser or client scripts (e.g. based on low-level tools from gitlab)

Examples:

- Simple listings or metadata download by browser (or curl)
- Advanced search with low-level tools
  (web interface to be reactivated and extended)

Download Configurations

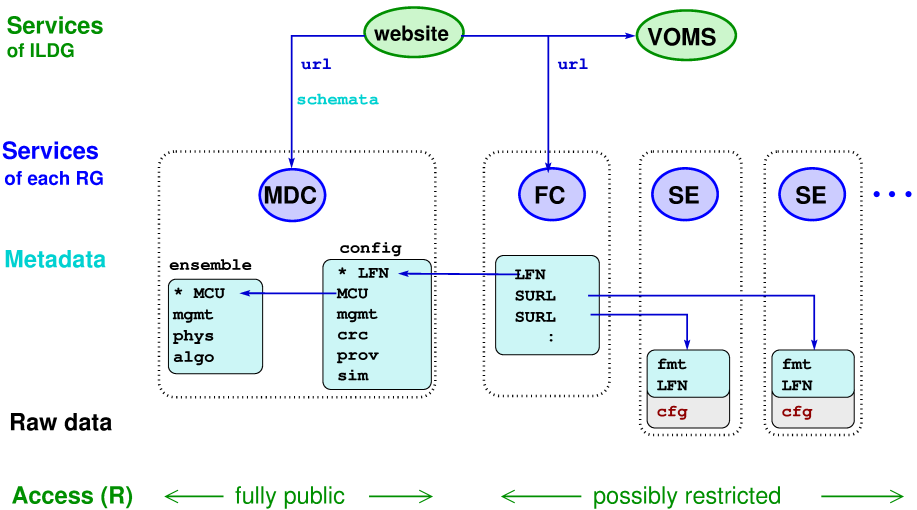
Prerequisites:

- ❒ Grid certificate and VO membership
- ❒ Client tools and scripts (e.g. from container used for hands-on)

Examples:

- Look-up of storage locations in FC
- Download by standard clients (e.g. `gfal-copy` or `globus-url-copy`)

Recap: Interplay between ILDG Services



Upload Configurations: prerequisites

- ❒ Grid certificate and VO membership
- ❒ Client tools and scripts (e.g. from container used for hands-on)
- ❒ Regional Grid resources, e.g.
  - storage space agreed with RG admins
  - RG-specific (naming) conventions and procedures
- ❒ ILDG-aware data management: required information must be available!
  e.g. parameters, log-files, provenance info, documentation (glossaries, references)
  also: well-defined license and usage policies, naming conventions, etc.
- ❒ Collaboration-specific tools
  e.g. simple scripts to extract data and metadata from production workflow

Upload Configurations: preparation

- Simple collaboration-specific scripts (`my_*`) to:
  - collect or extract information
  - generate markup (e.g. templates + generic pre-processing)
  - convert configs (+ generic packing tools)
- Decide identifiers, storage locations, etc.
- Request project manager to set up permissions (if needed)
- Set up automatized workflow (incl. generation of identifiers and checks)

Example:

```
m=‘my_get_mcu‘                                          # determine MCU
for n in ‘seq $nmin $nmax‘
do
  l=‘my_get_lfn $n‘                                     # determine LFN
  my_get_cfg       $n  -o /tmp/$n.raw                   # convert config
  ildg-binary ...  $l /tmp/$n.raw -o /tmp/$n.lime       # pack config
  my_get_xml       $l $m $n > /tmp/$n.xml               # generate config XML
done
```

Upload Configurations: execution

- ❒ Upload ensemble metadata to MDC
- ❒ For each config:
  - upload metadata to MDC
  - register in FC
  - upload data to storage element

Example:

```
m=‘my_get_mcu‘                                  # determine MCU
for n in ‘seq $nmin $nmax‘
do
  l=‘my_get_lfn $n‘                             # determine LFN
  s=‘my_get_surl $n‘                            # determine SURL
  try-mdc -ic /tmp/$n.xml                       # upload XML
  try-fc  -i  aca=$m  lfn=$l  surl=$s           # register in FC
  gfal-copy   /tmp/$n.lime  $s                  # upload binary
done
```