

# Workflow of ILDG Upload



ILDG Working Groups

July, 2025

# Overview

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

# Search and Download Metadata

## Prerequisites:

- ❑ Web browser or client tools (e.g. [low-level scripts from gitlab](#))

## Examples:

- Simple listings or metadata download by browser (or `curl`)
- Advanced search with low-level tools or web interfaces

# Download Configurations

## Prerequisites:

- ❑ VO membership (registration at [IAM](#))
- ❑ Clients tools to get token and access storage  
(e.g. `curl` or `oidc-agent` or `gfal-copy` or [low-level scripts from gitlab](#))

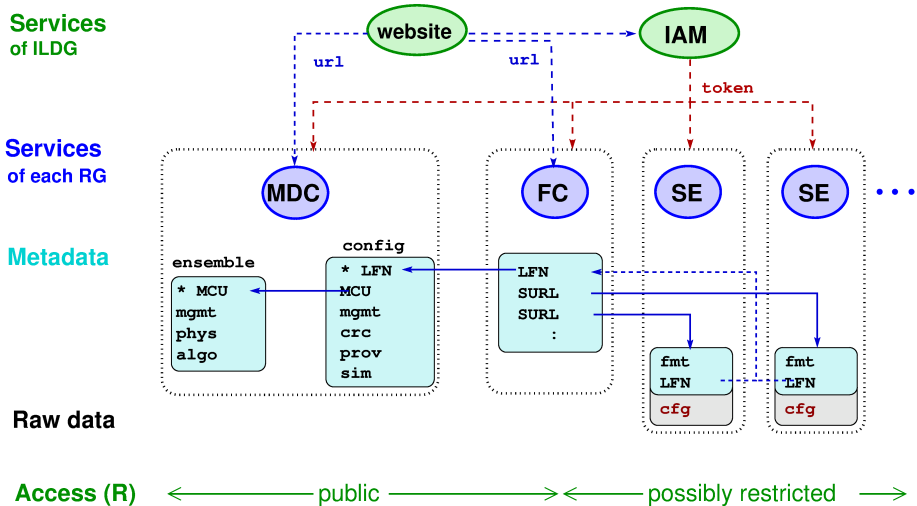
## Examples:

- Look-up of storage locations in FC
- Download by standard clients

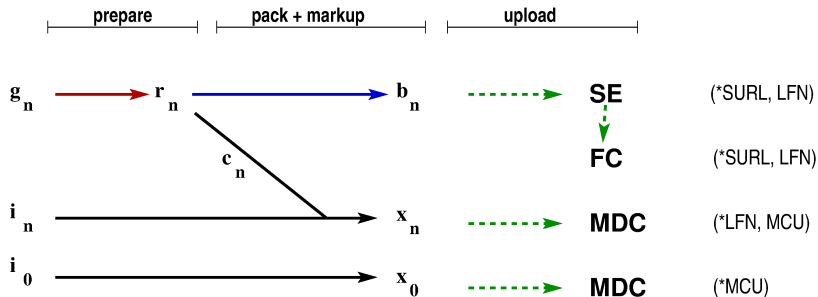
# Upload Configurations: prerequisites

- ❑ VO membership (registration at [IAM](#))
- ❑ Client tools and scripts to get token and write to catalogues and storage
- ❑ Regional Grid resources, e.g.
  - storage space agreed with RG admins
  - RG-specific (naming-) conventions and procedures
- ❑ ILDG-aware data management: information required for markup must exist somewhere!  
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

## Recap: Interplay between ILDG Services



# Upload Configurations: dependencies



Unique identifiers: MCU = markovChainURI, LFN = dataLFN, SURL = storage URL

$g_n$  = original gauge configs

$i_n$  = simulation info (for ensemble and configs)

$r_n$  = gauge configs converted to layout and format of ildg-binary-data record

$b_n$  = gauge configs packed into ILDG binary format (LIME)

$c_n$  = checksums of each config (crc, plaquette)

$x_n$  = XML metadata (for ensemble and configs)

# Prepare Configurations (binary data and XML)

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

## Example:

```
mcu=`my-get-mcu`                                # determine MCU
for n in `seq $nmin $nmax`
do
  lfn=`my-get-lfn $n`                            # determine LFN
  my-get-cfg-binary $n -o $tmp_blob              # convert config
  ildg-binary ... -LFN $lfn -i $tmp_blob -o $tmp_lime$n # pack config
  my-get-cfg-xml $lfn $mcu $n > $tmp_xml$n       # generate XML
done
```

# Upload Configurations

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

## Example:

```
for n in `seq $nmin $nmax`  
do  
  lfn=`my-get-lfn $n`           # determine LFN  
  surl=`my-get-surl $n`        # determine SURL  
  try-mdc -ic $tmp_xml$n       # upload XML  
  try-fc -i lfn=$l surl=$surl  # register in FC  
  try-se -p $tmp_lime$n $surl  # upload binary  
done
```

# Links

- ❑ ILDG Web Page: <https://hpc.desy.de/ildg/>
  - Specifications (QCDml, file format, API)
  - Services (URLs)
  - Organization (Working Groups, Meetings)
- ❑ Material from Hands-on Workshop July 2025
  - [instructions and presentations](#)
  - [material from exercises](#)
  - simple [client](#) and [packing](#) tools
  - [container environment](#)
- ❑ Metadata schemas: <https://gitlab.desy.de/ildg/mdwg/qcdml>
- ❑ File format: <https://gitlab.desy.de/ildg/mdwg/file-format>
- ❑ Simple MDC and FC listings, e.g. [https://hpc.desy.de/ildg/metadata\\_indices/](https://hpc.desy.de/ildg/metadata_indices/)
- ❑ Contact email (Working Groups, etc.): [ildg-contact\(at\)desy.de](mailto:ildg-contact@desy.de)