

ILDG Services and Middleware



Working Groups

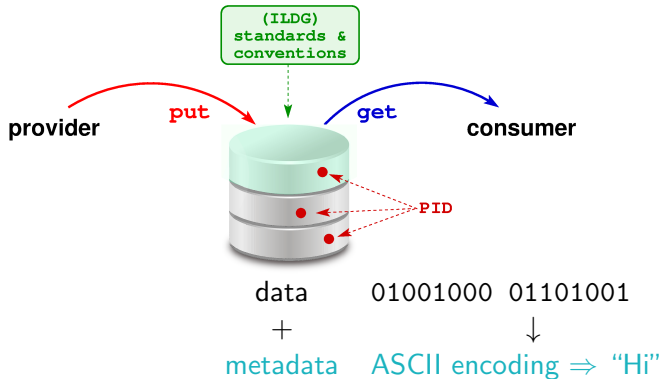
Hands-on Workshop

July 8, 2025

Overview

1. Middleware
2. Distributed Web Services
3. Basic ILDG Services
4. Interacting with ILDG Services

Naive (Meta-) Data Sharing

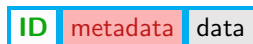


- (Meta-)data objects must have **persistant (globally) unique identifier(s)** **F1, A1**
- Many kinds of **metadata** (MD): format, content, provenance, access policies, ... **F2, R1**
- **Standards** and (community-specific) **conventions** **A1, R1.3**

FAIR data: a **local** implementation

Logical organization as a **database**:

- ❑ each FAIR data object becomes entry (row) in a database (table) with 3 fields (columns)



→ **retrievable by ID** (A1) and **searchable by metadata** (F4)

- ❑ mint **persistent identifiers** (ID)
- ❑ define appropriate **metadata schema** and storage format
- ❑ possibly implemented just through a local POSIX file system

i.e. buy a big disk and use standard tools: `ls`, `grep`, `find`, , ...

- ❑ additional **access control** mechanisms are required for “sharing” of data

N.B.: Technically, ILDG could be implemented by a **single central** infrastructure

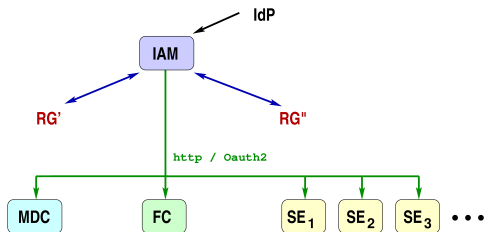
FAIR data: a **distributed** implementation

For **large** data objects or volumes: need to split (physical) storage of metadata and data



→ **distributed** implementation: typically by distinct **web services** (not pages):

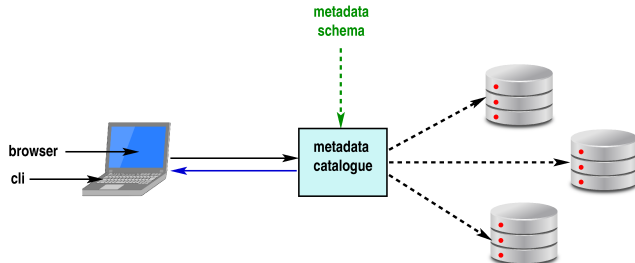
- Authentication and Authorization (AAI)
- Metadata Catalogue (MDC)
- File Catalogue (FC)
- Storage Elements (SE)



where

- separate MDC and SE becomes mandatory for **large data objects** (cost of search)
- multiple SE may become mandatory in practice (replication, funding, ownership)
- FC becomes mandatory if there are multiple SE or varying storage locations (SURL)

Distributed Interoperable Web Services



Not (just) web pages!

- grid (or cloud) storage elements (SE)
- central Metadata Catalogue(s) (MDC)

Global Services and Organization of ILDG

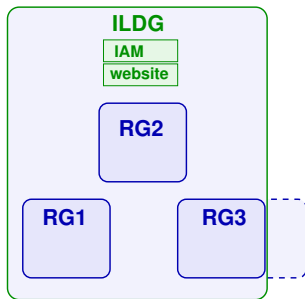
ILDG operates only **2 global services**:

❑ User registration (IAM)

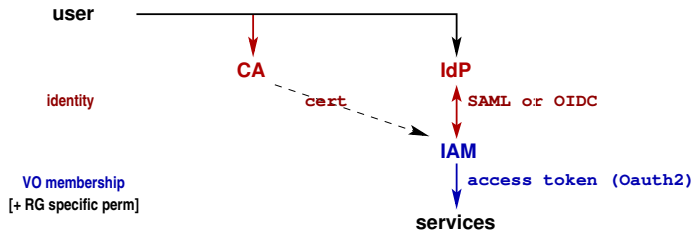
- registry of ILDG users (groups and policies)
- used for authentication to regional services

❑ Web page (under re-construction)

- specification of standards and conventions
- URLs of services of each regional grid
- VO Policy
- working groups and board



User Registration / Identity and Access (IAM)



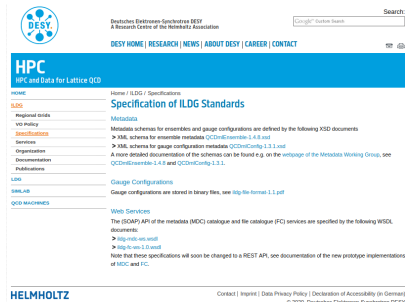
Now: (ILDG 2.0)

- identity = OIDC token or SAML from **trusted IdPs** (e.g. home institutions in **eduGAIN**)
- membership = **IAM** (Identity and Access Management) hosted at CNAF/INFN
- transport = access token (1 h)
- optional use of certificate only for convenient login

Currently 70 ILDG members with 30 different IdPs registered in IAM

Under (re-)construction, also URL may move

- Policies
 - AUP: SPs ↔ users
 - VO Policy: users ↔ users
- Specifications:
 - metadata schema
 - file formats
 - URLs of services (`Services.xml`)
- Working groups and Board
- (Incomplete) user documentation



<https://hpc.desy.de/ildg>

Autonomous Regional Grids

Services operated by each Regional Grid

- Metadata Catalog (MDC)
- File Catalog (FC)
- Storage Elements (SE)
- Website with RG-specific information

Regional Grids: CSSM, JLDG, LDG, UKQCD, USQCD

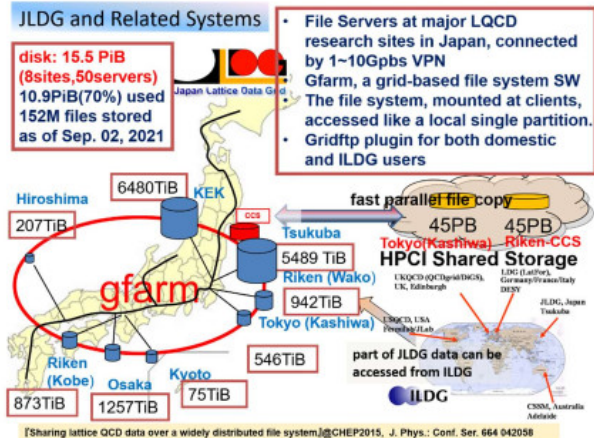
- are implemented with different architectures and technologies
- operate in an autonomous way with individual policies

Examples

- JLDG (Japan): single SE, no specific access control
- LDG (Europe): multiple SEs, fine-grained access control

JLDG Architecture

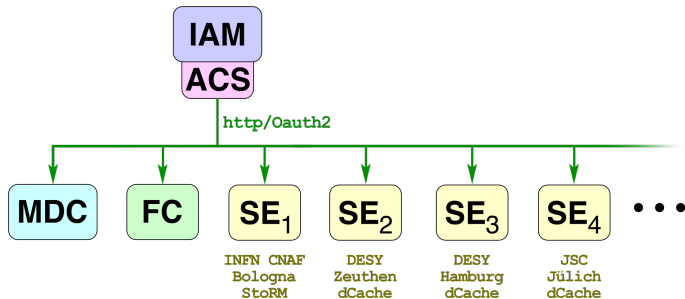
- Single federated storage system (GFARM)
- JLDG-internal write access
- Fast read access (gridftp) available for VO members
- Transition to token-based authentication in progress
- Prototype setup for data publishing and DOI minting
- Tightly organized as a Regional Grid



T. Yoshie

LDG Architecture

- Multiple SEs (4 at this moment)
- Use of WLCG; Partly “piggy-backing” on WLCG infrastructure
- Fine-grained access control via ACS



Metadata Catalogue (MDC)

Key purpose: ID registration and metadata search

ID $\longleftrightarrow$ metadata

Basic operations

- * **Search:** query $\rightarrow$ list of IDs (supporting powerful Xpath queries)
- * **Retrieve:** ID $\rightarrow$ MD (QCDml schema)
- Validate, insert, update, delete, ...

F4

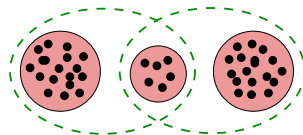
A1

Database schema

	metadata collection		
	config	ensemble	publication ...
primary key	LFN (dataLFN)	MCU (markovChainURI)	DOI
attributes	QCDml tree MCU	QCDml tree license	t.b.d. list of MCU DataCite metadata

ILDG Identifiers

- **Ensembles:** markovChainURI
`mc://<rg>/<collab>/<proj>/...`
- **Configurations:** dataLFN (no need of file extension)
`lfn://<rg>/<collab>/<proj>/...`



ID	entity	relation	content	data storage	access control
lfn	config	mc	yes	yes	no
	↓				↑
mc	ensemble	—	yes	no	yes
	↑↑↑				
DOI*)	publication	set of mc	yes	no	

*) ILDG 2.0 has no registration of IDs (DOI) or metadata for publications yet!

File Catalogue (FC)

Provides: functional (many-to-one) relation

FC: SURL $\longrightarrow$ LFN

Basic operations

- * list entries (SURL) by LFN
- list by other criteria (SURL, Access Control Attributes)
- insert, update, delete, ...

A1

Database schema

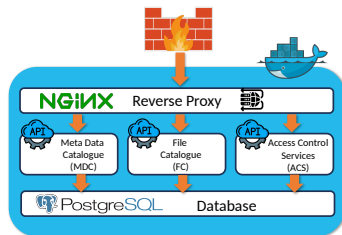
primary key: SURL (Storage URL)
attributes: LFN
MCU (or other optional Access Control Attributes)

New Implementation of MDC and FC

Technical details

- configurable, e.g. for additional collections (beyond ensembles and configs)
- additional attribute service for access control (ACS)
- REST API (see online documentation [MDC](#), [FC](#), and [ACS](#))
- simple deployment (Docker containers, Kubernetes in preparation)
e.g. for other regional grids or applications – currently JLDG, UK, and LDG (2 instances)

(by Basavaraja BS @ DESY/NIC)

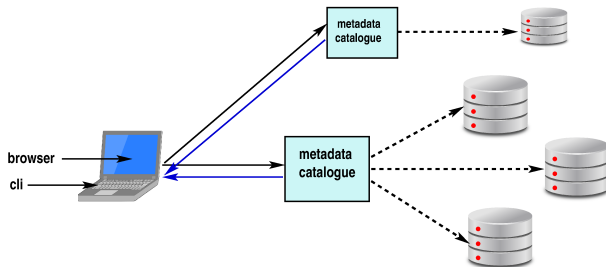


	MDC	FC	SE	
JLDG	✓	✓	1	gfarm access in progress
LDG	✓	✓	4	
UK	✓	✓	-	no legacy data

Optional add-on Services

- ☐ Markup GUI
- ☐ Query GUI (parametric in XSD)
- ☐ Metadata harvesting (OAI-PMH)
- ☐ File Transfer Service (FTS) with web monitor
- ☐ Fine-grained access control
- ☐ Monitoring
- ☐ Publishing

Interaction with ILDG Services



- ❑ Catalogues of all regional grids are **interoperable** due to standardized API
- ❑ High-level user operations may need several (≤ 10) low-level requests (e.g. HTTP) but still few compared to other web pages (implicitly handled by your browser)
 - www.google.com: $O(25)$ requests
 - www.github.com: $O(100)$ requests (try Ctrl-Shift-I in firefox)
 - your favourite airline: $O(200)$ requests

Use Cases and “ltools”

Consumer (collaboration internal)

- **lfind**: search in metadata catalog
- **lget**: download data and metadata



Consumer (community wide)

- optionally also use common search engines
- cite DOIs when using published data

Provider (collaboration internal)

- **lpack**: generate markup^{*)} and pack data
- **linit**: register ensemble metadata
- **lput**: upload config data and metadata



Provider (community wide)

- optionally register DOI and generate landing page
 - drop access restriction flag
- benefit from data citations

^{*)} trivial if information is already collected during production!

Examples of Interactions with ILDG Services

❑ “Single Sign-On” (SSO) to ILDG

- authenticate with your IdP (or certificate) $\longleftrightarrow$ IAM, IdP
- request group memberships $\longleftrightarrow$ IAM
- authorize tokens (configure clients) $\longleftrightarrow$ IAM

❑ “Get” config data (for specific and known LFN)

- optionally download config (and ensemble) metadata $\longleftrightarrow$ MDC
- determine SURL list $\longleftrightarrow$ FC
- get token $\longleftrightarrow$ IAM
- download data $\longleftrightarrow$ SE

❑ “Put” config data (for existing ensemble)

- decide SE and agree SURL $\longleftrightarrow$ RG admin
- get token $\longleftrightarrow$ IAM
- upload config metadata $\longleftrightarrow$ MDC
- register SURL $\longleftrightarrow$ FC
- upload binary $\longleftrightarrow$ SE

Consistency of ILDG Data

- ❑ ILDG is logically a distributed relational database with 2 kinds of entities

- configs: metadata + (binary) data
- ensembles: metadata

and corresponding primary keys

- LFN (dataLFN): *lfn://rg/collaboration/project/name*
- MCU (markovChainURI): *mc://rg/collaboration/project*

- ☞ Persistence and globally unique identifiers needs to be guaranteed by data providers.

- ❑ Typical inconsistencies (RDB anomalies) may arise from

- failures of individual services
- incorrect use of low-level tools

and can only be

- detected and fixed by regular scans (with possibly prohibitive cost)
- checked and handled by high-level tools (including roll-back)

Hands-on Exercises

Please keep in mind:

- ❑ you are using a prototype system, some components of which have been re-activated or newly developed only during the last months
 - ❑ currently we do not (yet/any more) have user-friendly “ltools”, but only quick and dirty scripts for low-level operations:
 - `try-mdc`, `try-fc`, `try-se` (just wrapper scripts for `curl`)
 - `lime-ls`, `lime-unpack`, `lime-pack1`
 - `ildg-cksum`, `ildg-binary`
- ... you all can improve them and contribute to develop proper user tools!

Hands-on Exercises (cont.)

- ❑ Optional SW packages are installed in the workshop image (docker + apptainer) which you can running according to the [instructions](#) .

In particular, the container includes

- curl other utilities
- gfal commands ([Grid File Access Library](#))
- oidc-agent ([documentation](#))

- ❑ Additional material is on [gitlab](#)

- [exercise instructions](#)
- [scripts for low-level testing \(try-*\)](#)

- ❑ Ready to get hands on? (and fingers dirty!)