

# Lattice QCD Data and Metadata Archives at Fermilab and the International Lattice Data Grid

Eric H. Nielsen, Jr.  
James Simone

CHEP '04  
September 29, 2004

# Lattice QCD

Lattice Quantum Chromodynamics is the numerical simulation of the strong interaction between quarks using discretized space and time.

Lattice QCD projects generate sets of large files representing gluon states through a sequence of times.

# The Opportunity

- Several groups generate and archive lattice QCD data.
- These data are vital resources both for other groups and for future use by the original authors.

# Challenges

- Scientists must discover and obtain data of interest for it to be useful.
- Significant metadata are required to interpret these data properly.
- The data volume can be inconveniently large; convenient use of mass storage systems is essential.

# The ILDG

The International Lattice Data Grid will provide infrastructure to help meet these challenges.

When complete, the ILDG will consist of an integrated set of clients and services, implemented (possibly independently) at each participating institution, that enables users to make data available to other scientists and to discover and use data made available by others.

# Functionality 1

- **export**: A user must be able to make data available.
- **explore**: A user should be able to interactively explore available data using an interface that presents an intuitive, hierarchical sorting of the data based on metadata.

# Functionality 2

- **discover**: A user must be able to list data whose metadata matches desired criteria.
- **query**: A user should be able to retrieve metadata (either specific parameters or full QCDml files) about a specific data file.
- **replicate**: a user must be able to copy a data file from one mass storage system into another when access to both is permitted.

# Functionality 3

- **get**: A user must be able to copy a data file and corresponding metadata onto local disk.
- **alter**: A user should be able to publish revisions to the metadata. (The data itself should not be alterable.)
- **revert**: It should be possible to revert to a previous version of the metadata in case of accidental corruption.



# Functionality 4

- **audit**: It should be possible to track usage based on a variety of parameters.
- **withdraw**: A user should be able to withdraw data from the ILDG. This operation will not make either the data or metadata unavailable, but it should prevent the file from being listed by metadata queries unless withdrawn data is specifically requested.

# Points of Collaboration

- The ILDG collaboration recently finished version 1.1 of a standard XML based metadata file format, QCDml.
- The ILDG middleware group is actively developing standard interfaces for the services used to support this functionality; these standards will allow clients implemented at one institution to use services implemented at others.

# User Interface

- Only the middleware interfaces will be standardized; user interfaces may vary by implementation.
- Details of the services and their uses can be hidden by a simple user interface designed to help users accomplish required tasks.
- User programs may contact services directly if it is more convenient.

# Interfaces

- The ILDG middleware working group is developing standard interfaces for these services using W3C (SOAP, WSDL) and globus (GSI) recommended standards.
- Use of standards will allow those implementing clients and services to take advantage of existing code libraries, and even enable ambitious users to use the services directly.

# Data Grid Services

The ILDG is following an architecture typical of data grids: users

- discover available services using a **web-service registry**,
- discover data and obtain necessary metadata using a **metadata database**,
- locate data using a **replica catalog**, and
- manage data movement using a **storage resource manager**.

# Sample Use for a Web-service Registry

- A client sends a SOAP message specifying whether metadata database or replica catalog services are desired.
- The web-service registry server sends the client a SOAP message listing URL's for known services of the desired type.

# Sample Use of a Metadata Database

- A client sends the metadata database server a SOAP message containing a query on a metadata database.
- The metadata database server processes the query and sends a SOAP message back to the client with the results of the query. The result can be the full QCDml, a list of global file names for the described data, or a specific metadata parameter.

# Sample Use of a Replica Catalog

- The client sends a SOAP message with a global file name to the replica catalog.
- The replica catalog returns a list of URI's with which the data may be obtained. The URI may be a traditional URL or the URI of a web service (such as an SRM) that will provide the data.



# ILDG-MWVG Status

- A requirements overview, an architecture document, and a set of base use cases have been submitted to the ILDG **M**iddleware **W**orking **G**roup.
- At a working group meeting planned for this fall, we plan to:
  - refine and reach consensus on these documents, and
  - generate an initial set of service interfaces.

# The Fermilab Prototype

- Participating institutions are experimenting with prototypes.
- The Fermilab prototype uses:
  - Python,
  - PostgreSQL,
  - Globus, and
  - pyGlobus.

PyGlobus provides a python interface to the Globus toolkit.

# Fermilab Prototype Status

- ✓ Fermilab SRM implementation provides storage resource management.
- ✓ Metadata database and replica catalog implement a basic set of essential features.
- ✓ The prototype command line client supports integrated use of the services.
- ✓ A web interface to the metadata catalog allows browsing of available data.
- × Few data sets currently have metadata in the proper format.

# Fermilab SRM

- The Fermilab SRM implementation was developed outside of the lattice QCD project, and is in production use by other projects.
- The Fermilab ILDG prototype client contacts the SRM server directly instead of using the stand-alone SRM client supplied.
- Transfer of lattice QCD data from made available at remote sites through a variety of methods (http, ftp, grid-ftp) has been demonstrated.

# Metadata Catalog Challenges

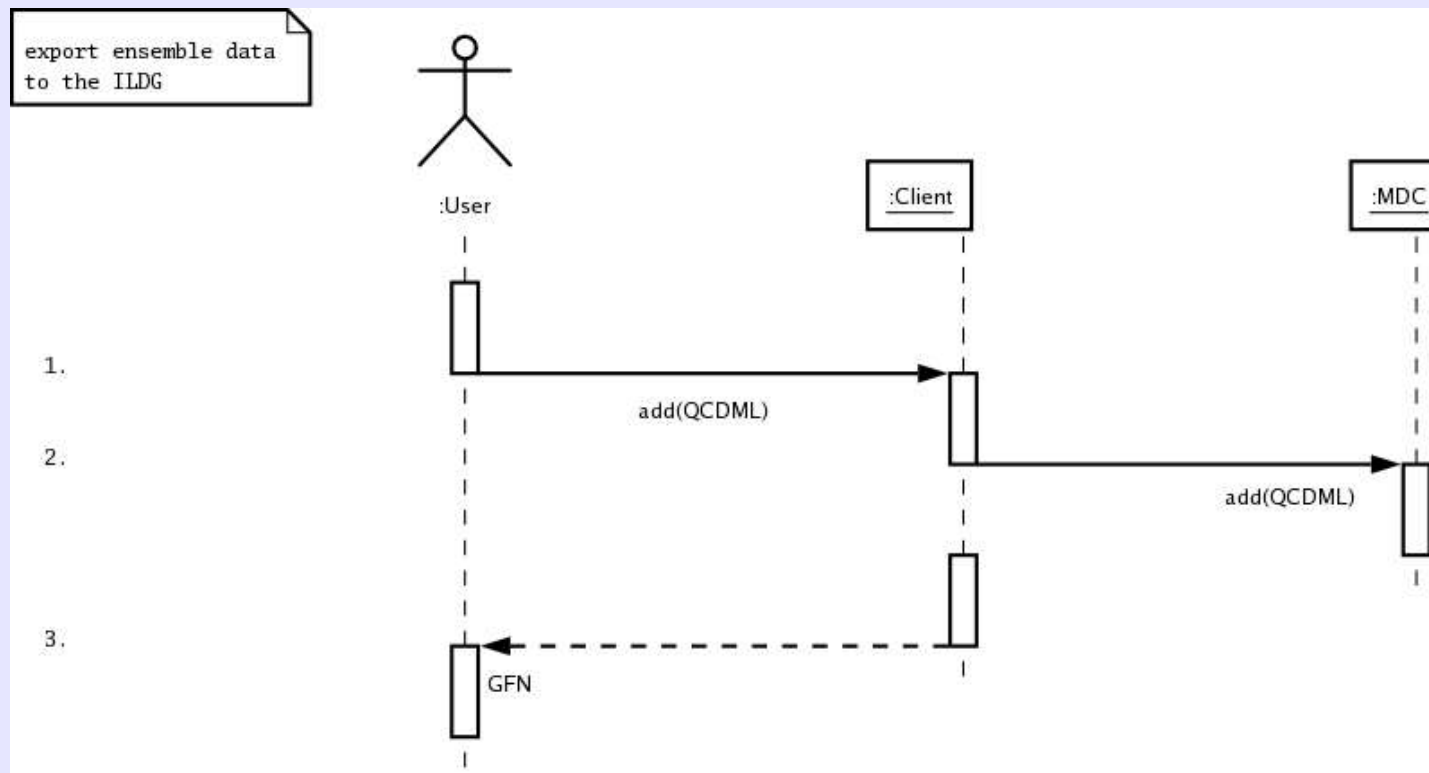
- Data naturally divided into sets:
  - Each file contains data on one configuration, each of which has a corresponding QCDml metadata file.
  - Each configuration is a member of an ensemble, itself described by a QCDml file.
- Native XML databases are awkward to query interactively and may not scale well.
- While naïve mappings onto a relational database are possible, they are not optimal; we prefer a designed database.

# Mapping QCDml Into a Database in the Fermilab Prototype

- The metadata catalog service fills a human designed set of database tables using values supplied in QCDml, and stores the QCDml itself as well.
- The service presents the data stored in the set of tables as a single SQL View, which is simpler for users to query than the raw tables or QCDml.
- In the future, queries may include Xpath expressions on the QCDml itself.

# Export Ensemble

```
bash$ ildg export-ensemble mydata1.qcdml
```



# Export Data

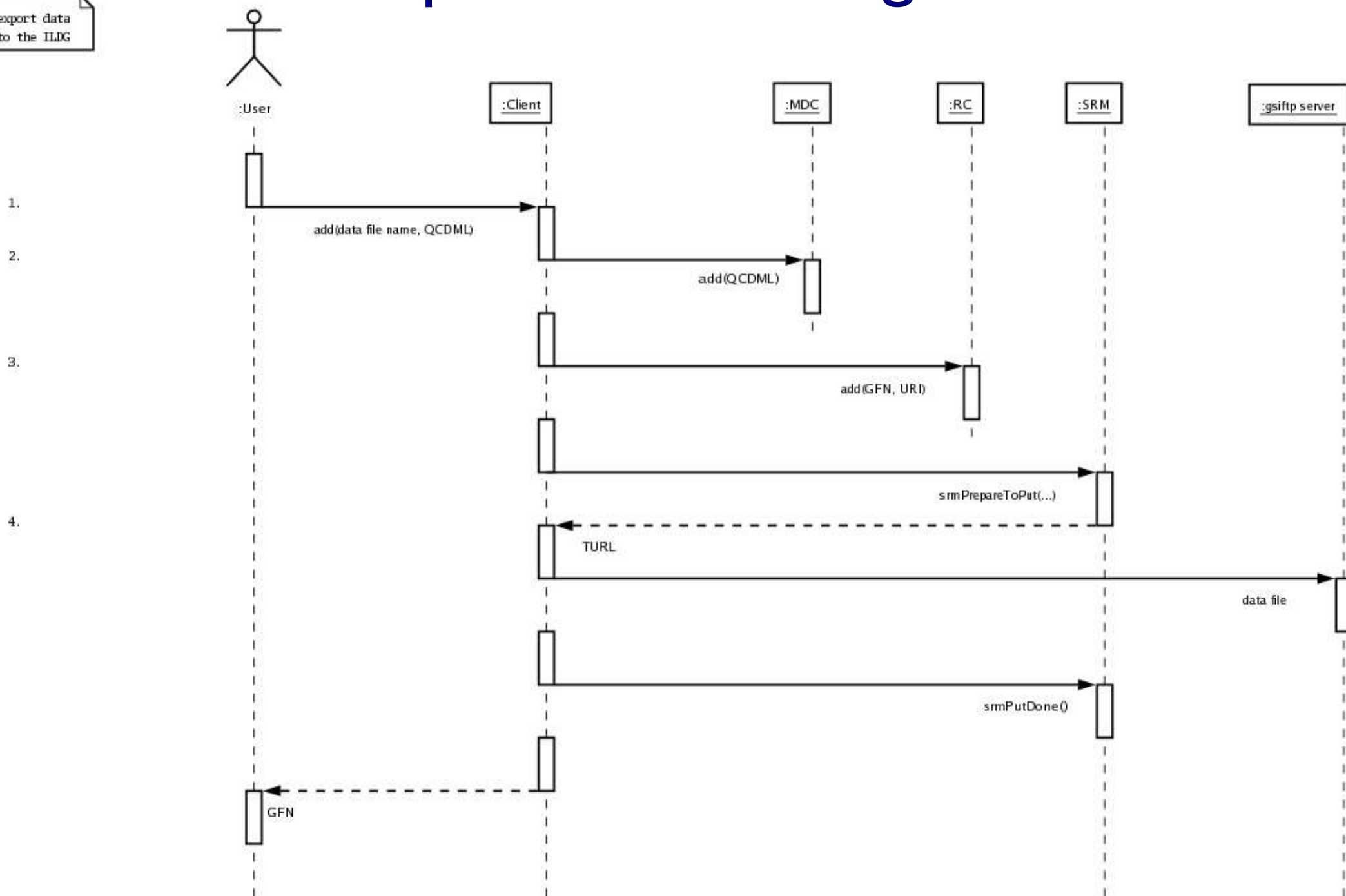
```
bash$ ildg add /data/myData.dat /data/myData.xml \  
> srm://myrobothost.mylab.org/myDirectory/myData.dat
```

Adds the metadata to the metadata database, copies the data file into the mass storage system, and updates the replica catalog with the location of the newly added file.



# Export Data Diagram

export data to the ILDG



# Discover

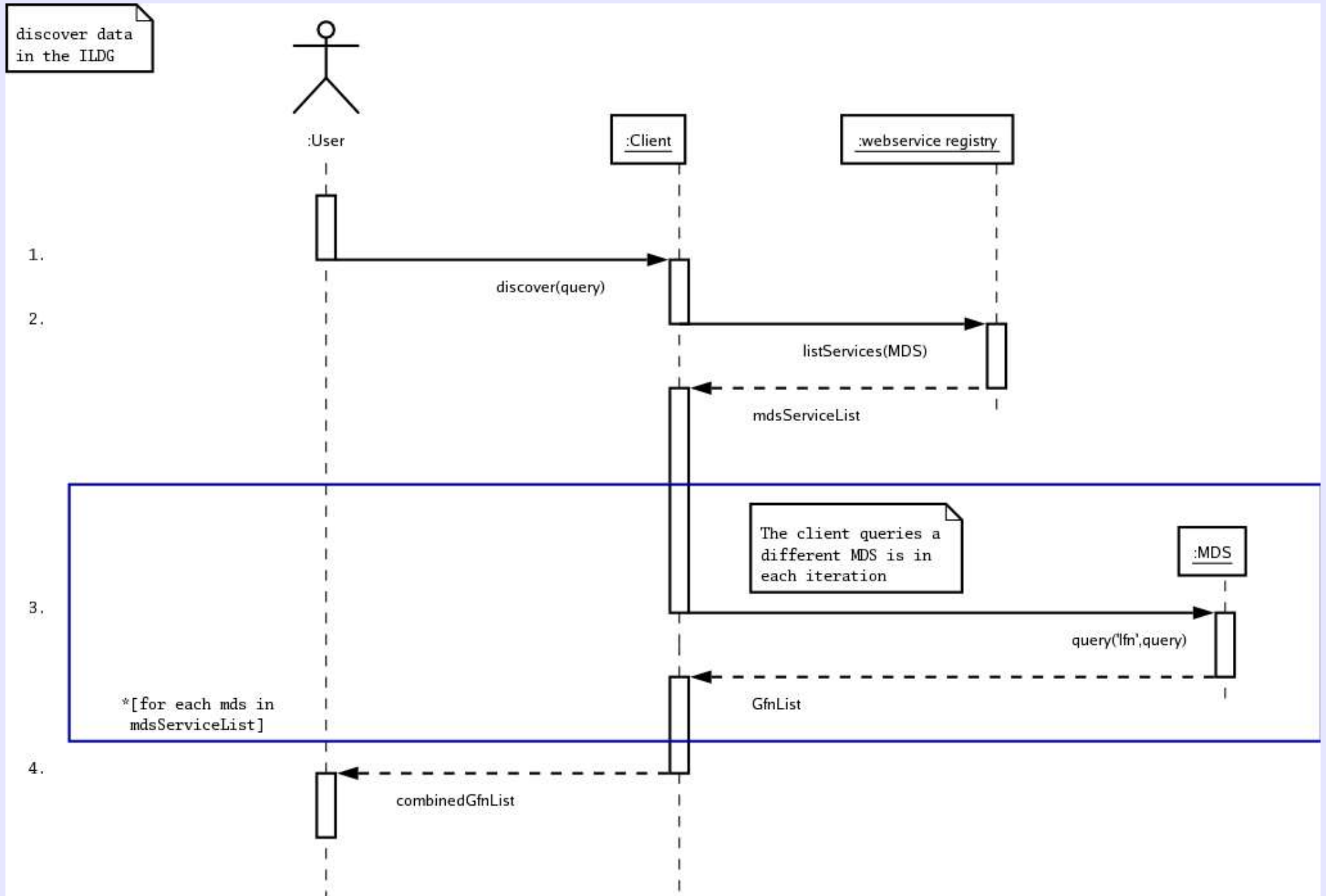
```
bash$ ildg discover \
```

```
> "institution = 'Fermilab' AND date = '2003-12-03'"
```

Queries available metadata databases for data whose metadata matches criteria expressed in terms of simple keywords. These keywords usually, but not always, correspond directly paths in QCDml.

The prototype also includes a web-browser based interactive search facility which allows users to avoid composing queries altogether. While the user must still understand the meaning of the various keywords to intelligently search the database, learning the keywords or QCDml XPath is not necessary.

# Discover Diagram



# Obtaining Data and Metadata

```
bash$ ildg get "projectName = 'myProject" \  
> '/data/%s-series%s-step%s.dat' \  
> 'projectName series updates'
```

```
bash$ ildg query "projectName = 'myProject" \  
> '/data/%s-series%s-step%s.xml' \  
> 'projectName series updates'
```

Copies all data meeting specified conditions and corresponding metadata into local files named according to metadata parameters. Note that the user never needs to know the location of the archives in which the data are stored or the file names used by these archives or even the ILDG services.

# Future Directions

- Generate metadata files in the newly standard format for existing data, and add this data to the prototype.
- Formally agree on details of the functionality.
- Refine and reach consensus on standards for interfaces.
- Implement the standards at each site.