



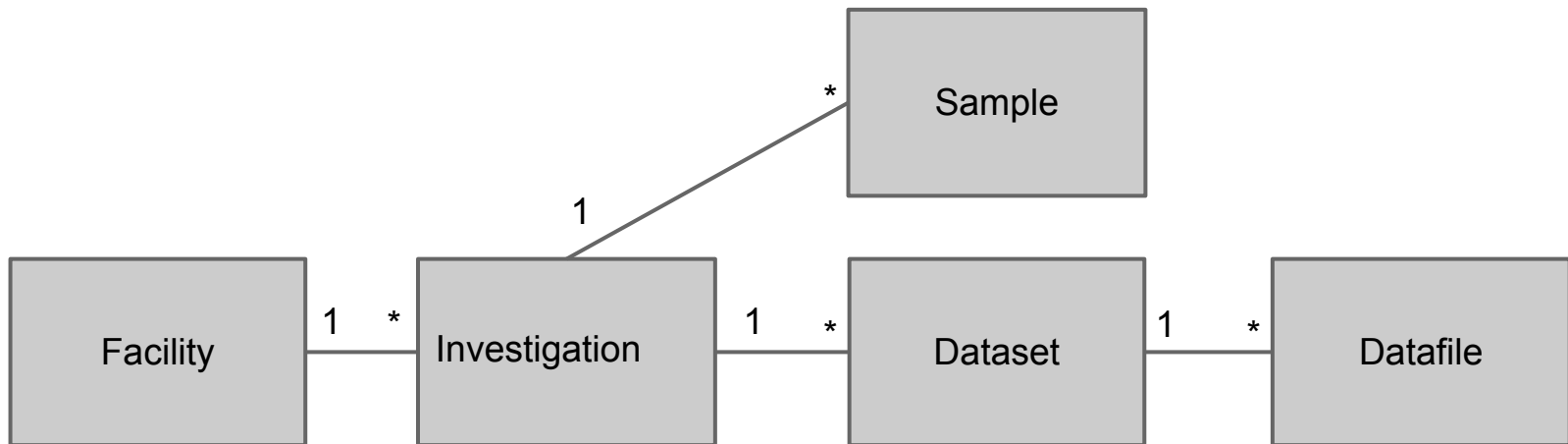
# ICAT

3 → 4.0 → 4.1

Status 20/3/2012

Steve Fisher

# Schema - main components



**These all take arbitrary parameters as an extensibility mechanism**

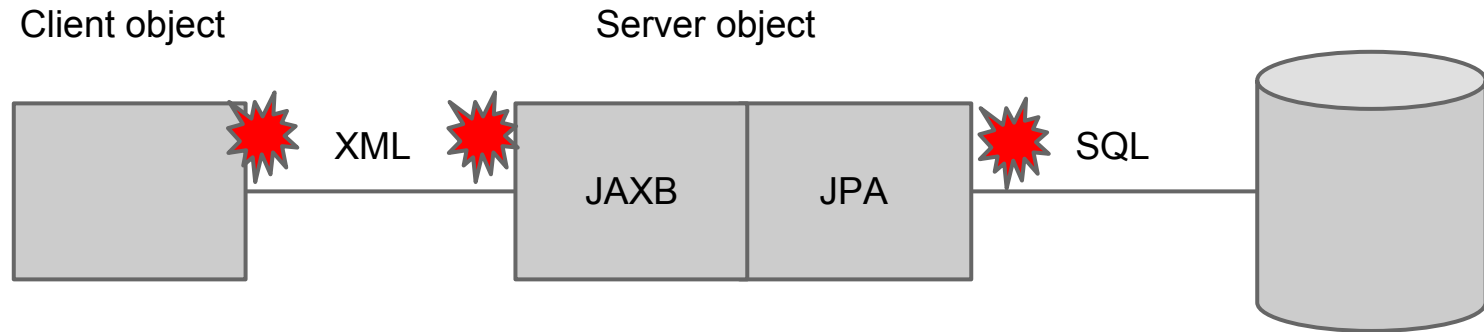
**Parameters added to investigation in release 4**

# Implementation

- RDBMS
  - Any supported by eclipselink (4.0)
  - No longer tied to Oracle (4.0)
- RDBMS accessed (DDL and DML) via JPA
- EJB in an EAR file deployed in an application server
  - e.g. glassfish
- Exposed as a web service
  - New very small API (4.0)



# Architecture



- JPA works very well and bridges the Java RDBMS with only a few compromises

- JAXB is however a major restriction as it only supports tree structures

# The new API (4.0)

- login/logout

- returns a `sessionId` to identify you for a finite period

- **C**reate

- Java e.g.

```
Investigation inv = new Investigation();  
inv.setName("Fred");
```

```
...
```

```
Long id = (Long)ep.create(sessionId, inv);
```

- **R**ead

- actually search and get

- **U**ppdate

- **D**elete

- There are no schema specific operations

- Schema can be modified without side effects

- "Create" can create a tree of objects in one call

# Getting started

```
URL hostUrl = new URL("http://localhost:8080")

URL icatUrl = new URL(hostUrl, "/ICATService/ICAT?wsdl");
QName qName = new QName("client.icat3.uk", "ICATService");
ICATService service = new ICATService(icatUrl, qName);
ICAT icat = service.getICATPort();

String sessionId = icat.login("userName", "password");
```

# A bit more on create ...

```
Facility f = new Facility();  
f.setName ("Some facility");  
f.setDaysUntilRelease(90);  
icat.create(sessionId, f);
```

```
Investigation inv = new Investigation();  
inv.setName("Fred");  
inv.setFacility(f);  
Dataset ds = new Dataset();  
ds.setName("A dataset");  
ds.setType(type);  
inv.getDatasets().add(ds);  
Long id = (Long) icat.create(sessionId, inv);  
inv.setId(id);
```

- Do ingestion this way.
- Build tree as defined by "cacade" and make one call.
- Don't set backwards reference

# update ...

```
String pk = "Some facility";  
Facility f = (Facility) icat.get(sessionId, "Facility", pk);  
f.setDaysUntilRelease(30);  
icat.update(sessionId, f);
```





# and delete ...

```
String pk = "Some facility";  
Facility f = (Facility) icat.get(sessionId, "Facility",  
pk);  
icat.delete(sessionId, f);
```

- Again, "cascades" are followed so this delete would have a major effect.



# get

```
String pk = "Some facility";  
Facility f = (Facility) icat.get(sessionId, "Facility", pk);  
  
Long dsid = 76347;  
Dataset ds = (Dataset) icat.get(sessionId, "Dataset", dsid);  
  
Dataset ds = (Dataset) icat.get(sessionId,  
"Dataset INCLUDE Datafile, DatasetParameter, DatafileParameter",  
dsid);
```

**ds.getType() will be null**

**Included entities must be linked via a graph without loops - i.e. only one route.**

# search

```
String query = "Dataset"  
List<?> results = icat.search(sessionId, query);  
  
query = "Dataset.name";  
  
query = "DISTINCT Dataset.name";  
  
query = "Dataset INCLUDE Datafile, DatafileParameter";  
  
query = "Dataset.id ORDER BY id"  
  
query = "MAX (Dataset.id) "  
  
query = "3,5 Dataset.id ORDER BY id"  
  
query = "3, Dataset.id ORDER BY id"  
  
query = " ,5 Dataset.id ORDER BY id"
```

# Search restrictions

```
"Dataset [type.name = 'GS' OR type.name = 'GQ']"
```

```
"Dataset [type.name IN ('GS', 'GQ')]"
```

```
"Dataset <-> DatasetParameter[type.name = 'TEMP' AND  
numericValue > 300]"
```

```
"Dataset <-> DatasetParameter[(type.name = 'TEMP' AND  
numericValue > 300) AND (type.name = 'PRESSURE' AND  
numericValue > 1020)]"
```

```
"Dataset <-> DatasetParameter[type.name = 'TEMP' AND  
numericValue > 300] AND [type.name = 'PRESSURE' AND  
numericValue > 1020]"
```

The restriction in the square brackets can be as complex as required - but must only refer to attributes of the object being restricted

Q .Why the strange query language?

A. To support authz

# Authz

- Uses four tables: User, Group, UserGroup and Rule
- Normal tables but special authz
  - users specified in a configuration file can create, update and delete
  - otherwise couldn't get started
- A rule authorizes a group member to **C**reate, **R**ead, **U**ppdate or **D**elate
  - By default access is denied to all objects
  - Rules allow access to be allowed - no deny mechanism
  - The rule is only applied to the object referenced directly in the API call.

<b>group</b>	<b>crudFlags</b>	<b>what</b>
user-office	"CRUD"	"Investigation"
null	"R"	"ParameterType"
FredReaders	"R"	"Datafile[name = 'fred']"
null	"R"	"Dataset <-> Investigation <-> InvestigationUser [user.name = :user]"

# Authz and INCLUDE

The rule applies to the first object or list of objects

Need to specify which related entities may be returned

Not yet implemented

Proposal:

- Recursively

- Allow following to-many if cascaded

- Allow following a to-one relationship but not subsequent cascades

Could have user specified rules  
but the generic rule appears  
sufficient

# Notifications

Uses search/authz syntax to control sending of JMS messages

<b>crudFlags</b>	<b>what</b>	<b>destType</b>	<b>dataTypes</b>	<b>name</b>	<b>...</b>
"R"	"Dataset"	PUBSUB	USERID, ENTITYKEY, NOTIFICATIONNAME	"DS"	
"CUD"	"ParameterType"	P2P	USERID, ENTITYKEY, NOTIFICATIONNAME	"PT"	
"R"	"Datafile[name='Fred']"	P2P	USERID, NOTIFICATIONNAME	"FR"	

**Does not send ICAT objects  
hence schema independent**

**Can be used as basis of general  
ICAT usage reporting mechanism**

# A note on error messages

- ICAT now makes only those checks that the RDBMS cannot check
  - This is much more efficient
- It catches exceptions thrown by the RDBMS and as these do not relate to client side concepts try to work out what went wrong and throw appropriate exception
- If exception not understood send it back to the user
- Unable to trap JAXB exceptions



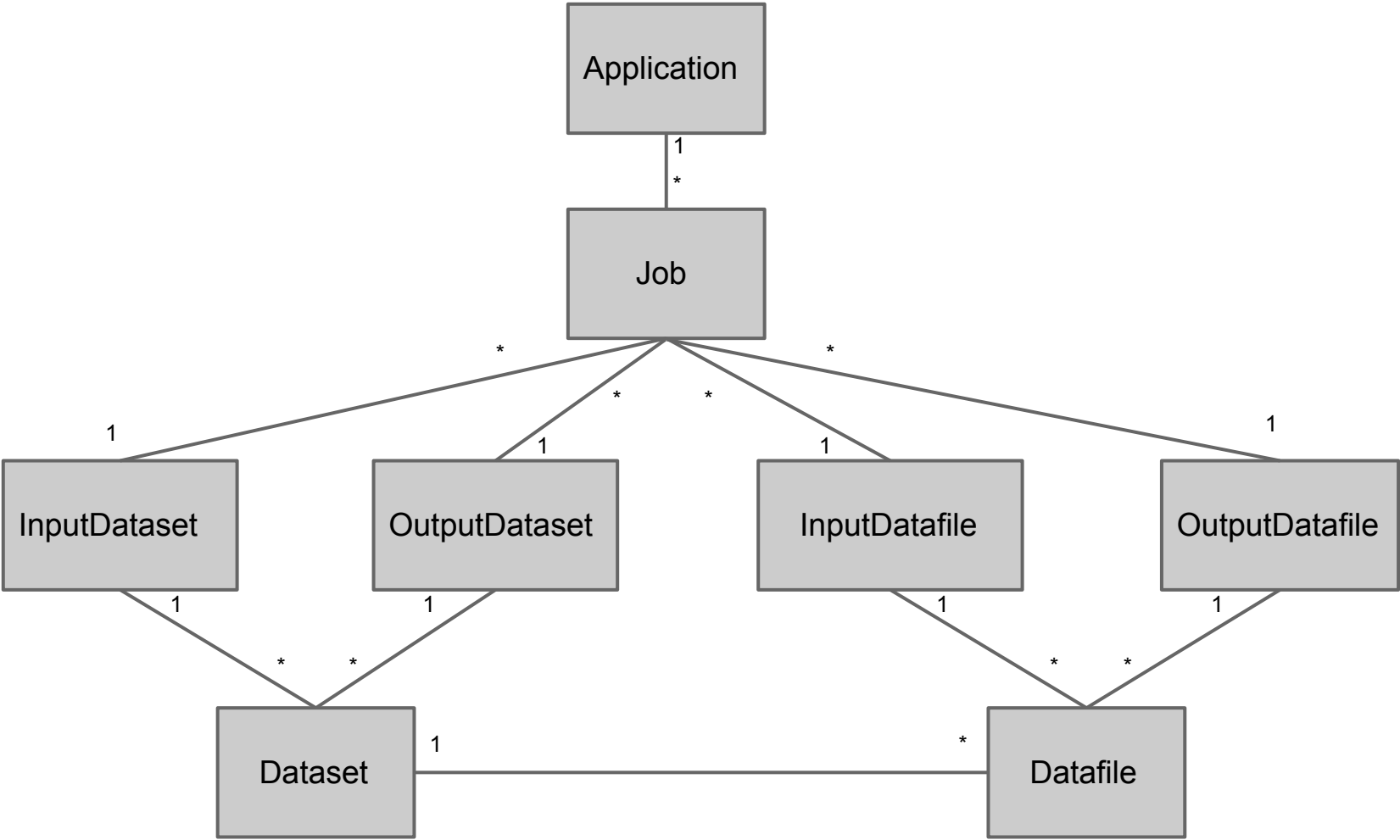
# Schema changes already in (4.1)

- Remove attributes that are not of general utility and use xxxParameters instead
  - some still present but marked with \*\*\*\* in the comments
- Those entities which represent many to many relationships now formed by concatenating the names of the related entities
  - “Investigator” has now become "InvestigationUser"
- All relationships may now be navigated in both directions
- Relationship fields follow normal convention
- Harmonised names of the various string fields
- Reduce number of non-optional fields
- Added notification request
- Added job etc. for provenance

# Proposed changes

- Need a ShiftUser (many to many)
  - otherwise shift is of little use.
- Permit mixing data from more than one facility
  - Particularly interesting for an institutional or personal ICAT
  - Requires that a number of unique constraints would need to include the facility. This would affect the various “Type” and “Status” entities.
- Add restrictions to ParameterTypes to define:
  - possible values for strings
  - upper and lower limits for numbers
  - boolean to indicate whether or not ICAT should respect the constraint
- Get rid of Topic
- Add DOI support
  - where does it go?
- <http://www.icatproject.org/icat.html>
  - generated from the icat code

# Simple representation of provenance



# New Generic Tools

These can now be written because of the regularity of the schema

- Admin Gui
  - There are some operations which are very infrequent - for example adding a new DatasetType.
- Data server (IDS2)
  - Makes special use of the location fields
- Data (and metadata) ingester
  - Use both direct calls to ICAT and some via IDS2



# Admin Gui / Editor

Goes through the API and therefore enforces the authorization rules and any additional constraints imposed by the server code.

It has been tailored to manipulate things which are not very plentiful - i.e. DatasetTypes rather than Datasets.

RE: ICAT x ICAT - G x Edit App x ICE x Issues to x LSF UBL x iGoogle x

hepntl131.esc.rl.ac.uk:8080/editor-1.0-SNAPSHOT/

Admin Phone Software R-GMA EMI Scott SRF LSF Travel Other Bookmarks

Create Facility Go Logged in as root Logout

description

name \* fred

fullName

daysUntilRelease \* 34

url

Submit

***With this editor and the new API there is no need to edit the DB manually***

<http://localhost:8080/editor-1.0-SNAPSHOT/>

# Data Server - IDS2

- Looks after the data as well as the metadata
- Uses read and write permissions to metadata to control access to data
- Makes use of the location fields in the Datafile and Dataset to locate individual files and sets of files.
- Assumes two "file systems" one fast and one for archive
- Archive holds zipped datasets - delayed write
- Can return a complete dataset or just one file identified by id or location
- "REST"ful web service provides easy access from Python, C++ etc
- See: <http://www.icatproject.org/ids2/apidocs/>

Should ICAT have a restful API?

Should IDS2 have a WSDL API?

# Data (and metadata) ingester

- A "trigger file" is dumped in a predefined directory after the data files are ready to be processed.
- Trigger file holds location of a directory
  - defines the dataset
- A program is invoked on the directory
  - generates XML to define dataset information and to identify which files should be and how
  - program provided by the facility
- Investigation is chosen based on dataset time and the instrument producing it
- Only 970 lines of code (excluding tests)

**Only a prototype but  
looks promising**

# What is left for 4.1

- Finalise the schema
- Finalise the API
  - suggested name org.icatproject.icat
- Internals
  - Authz for INCLUDE
  - More generic code
- Provide adequate error messages for common conditions
- Provide DB migration from ICAT3
- Separate off and restore the XMLIngest
- Restore the now separated off reporting system

## Opportunities to contribute:

Test on one or more non Oracle RDBMS

Provide DB migration (Oracle to Oracle) from ICAT3

Reporting system

Study performance - e.g. keeping Rule table in memory



# What might go into 4.2

- Provide old deprecated search calls - some are present in 4.1
  - Hope these are not needed - are they?
  - Could be contributed if anybody wants it
- Improved error messages for less common errors
- Better separation of Authn - probably chained
  - Should be Umbrella compatible
- "REST"ful API
  - This is probably difficult to do well



