# ICAT 4.3

Last chance to have your say

Steve Fisher / RAL

<dr.s.m.fisher@gmail.com>

# Internal Changes

- Deal with most of the TODOs
- Eliminate as much manually prepared code as possible by making more use of reflection
  - Reduces code volume
  - Simplifies maintenance

# Planned changes to ICAT 4.3

New operations:

*void refresh(String sid)* method

- This is useful for long running programs
- A program may be given a sessionId and keep on renewing it without storing the password

*void testCreate(String sid, Entity entity)*

*void testUpdate(String sid, Entity entity)*

*void testDelete(String sid, Entity entity)*

- simply throw the exception that the corresponding create, update or delete call would throw or do nothing
- useful for UIs to not offer operations that will fail

# Planned changes to ICAT 4.3

## Remove existing notification mechanism

It has a number of problems:

1. exposes information via JMS that can be used without making an ICAT call
2. requires an extra DB query after every operation
3. tricky to configure
4. can produce excessive JMS traffic

New mechanism will be described later

## Remove compat interface

It was planned to remove this backwards compatibility feature as soon as ICAT 4 was established

# Planned changes to ICAT 4.3

Change uniqueness constraints to:

- Datafile: dataset, name (-location)
- Dataset: investigation, name (-sample -type)
- Application: facility, name, version (+facility)
- Sample: investigation, name (-type)
- Investigation: facility, name, visitid (-facilityCycle)
- SampleType: facility, name, molecularFormula (+molecularFormula)

- Generally now container and name within container
- Application previously did not include facility
- Investigation includes visitid (unfortunately - as required by ISIS and DLS)

# Planned changes to ICAT 4.3

## Make the units of the ParameterType not nullable

- to support those databases that do not allow uniqueness constraints to include nullable fields.
- Change existing null values to the string value "None"

## Add a string attribute: “*arguments*” to the job to store the program arguments that were used.

- Should be a harmless schema change

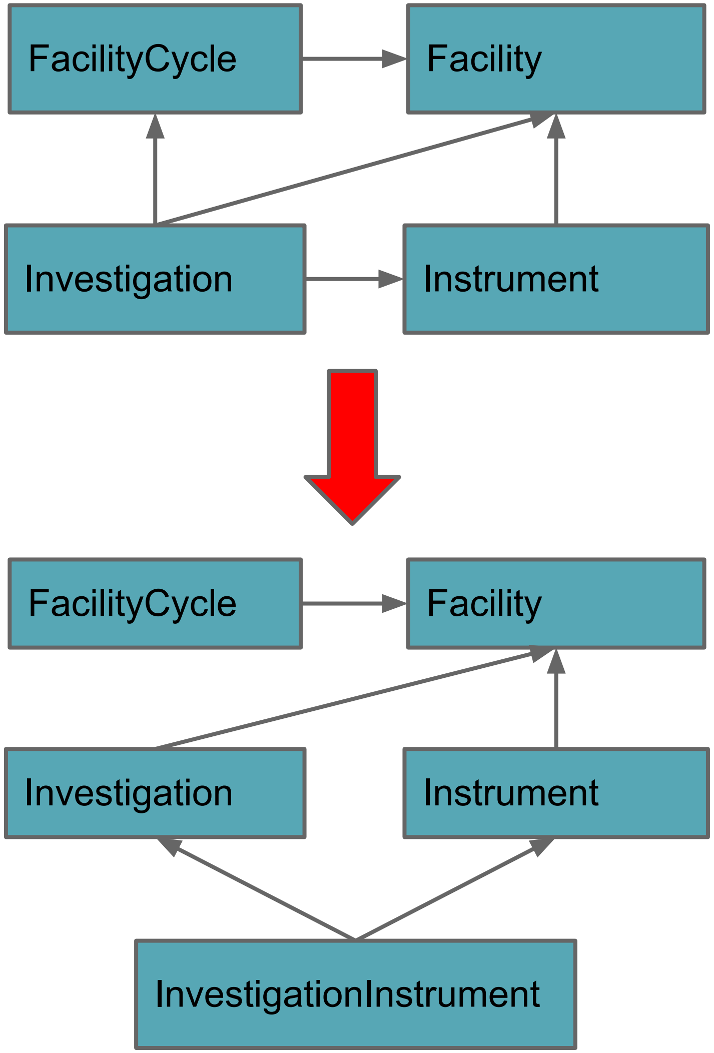
# Planned changes to ICAT 4.3

Eliminate the relationship between FacilityCycle and Investigation

- avoids a loop for better normalization
- may impact users

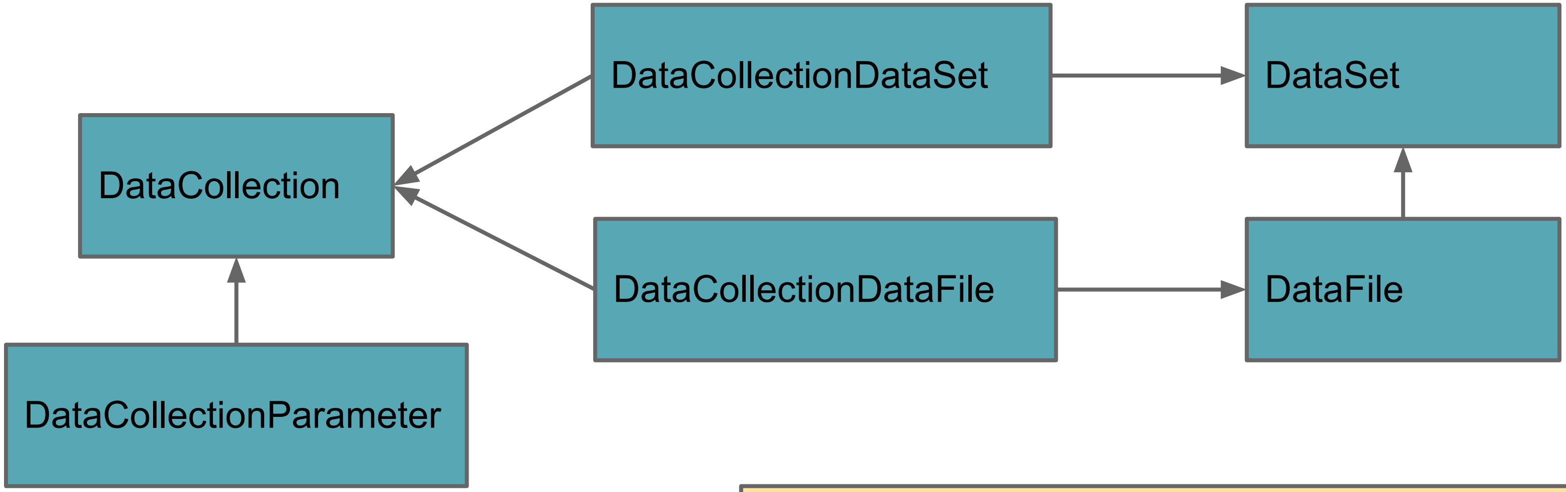
Add a table between Investigation and Instrument

- to represent many-to-many relationship between them
- this may impact users as an investigation currently makes use of at most one instrument.



# Planned changes to ICAT 4.3

Add a DataCollection object as a set of Datafiles and Datasets - with parameters



Brian prefers a Run entity – as an analogue to the Job entity, and both being specialisations of the W3C Prov “Activity” entity http://www.w3.org/TR/prov-primer/ (or the OPM “Process” entity - http://openprovenance.org/ ).

- From my understanding DataCollection can do the job of a "run"
- Parameter provides extensibility as usual

# Planned changes to ICAT 4.3

## Provision of conversion scripts

- Conversion scripts will be written for Oracle to directly change the schema and data
- Scripts will also be provided to identify problems with the uniqueness constraints as Oracle will only report that the constraint cannot be applied not where the problem is.
- Are they needed for MySQL?

# Rejected changes to ICAT 4.3

## Replace test.py by a Java equivalent

- No interest expressed by any facility

## Use new DataCollection for provenance

- Nice to say that a job is related to an input DataCollection, an output DataCollection and an Application. However:
  - it would break user code
  - current query language is not able to cope with two relationships between objects of the same type
  - will be reconsidered when/if the query language is enhanced

# Speculative features in ICAT 4.3

- A speculative feature should not be relied upon to be present in the next release
  - It may be changed in an incompatible manner
  - It may be dropped

- Three features proposed:
  - Call logging
  - New notification mechanism
  - Free text search

Call logging and the new notification mechanism replace the old notification system and should satisfy the notification use cases

All three features should be rather straightforward to provide and will require only very small changes to the established installation procedures.

# Call Logging

- New entry in icat.properties with a name of log and possible values: *none, file, tables or both*
- Three tables used for logging:
  - LogAccess (for login and logout): *user, sessionId, operation, timeStamp and duration*
  - LogWrite (for create, createMany, testCreate, update, testUpdate, delete, deleteMany or testDelete ): as LogAccess plus *entityName and entityId*
  - LogRead table (for get and search): as LogWrite plus the *query*
- The operation takes values: *login, logout, get, search, create, createMany, testCreate, update, testUpdate, delete, deleteMany or testDelete*
- For createMany, deleteMany and search, the information will be for the first table in the list.

# Call Logging - continued

- Logs contain potentially sensitive information
  - file access controlled by OS
  - tables by ICAT Authz

- Text file will be as compact as I can make it - probably one line per entry with tabs between fields.
  - The query field if present will go at the end as it can in principle contain tabs

- No facilities for managing the log output

# Notification

- Two especially interesting use cases to support automation are:
  - the arrival of a new Datafile
  - the marking of a Dataset as complete

- Generate JMS notification for every datafile or dataset which is created, updated or deleted

- icat.properties gets new entry to control the six publishing cases
  - publish: create datafile update dataset

- The message just holds:
  - entity name, (datafile or dataset) - in header
  - operation (create, update or delete) - in header
  - entity id - in body

- A client can use the header fields for selection.

# Notification - notes

- In the case of createMany or deleteMany many notifications will be sent.

- Constant, but moderate, level of JMS traffic

- No sensitive information exposed - the receiver needs to do a search or get, subject to the Authz rules

- Receiver:
  - should make use of header to only receive possibly relevant messages
  - should normally do *search* with a condition on the id rather than *get*
  - might batch up searches

# Free text search

- Use the Lucene indexing engine (which is used by Solr)
- Index updated by all create, update and delete calls.
- All text fields will be indexed but only the entity name and entity id will be returnable - and even then they will not be directly accessible but only via a new ICAT call:

  *List<Object> searchText(String sid, String luceneQuery)*

- ICAT authz rules will check that you are allowed to see the object before including it (without INCLUDES) in the list

# Free text search - details

- Lucene uses "fields" to provide context for a search.
- Index everything twice, once with a field of the entity type and once with a default field
- Lucene supports multiple query parsers. The "classic" Lucene parser includes support for fuzzy, proximity and range searches, boolean operators and grouping
- Build a single string for each entity with concatenation of all text fields (space separated) could also include date fields (yes?) and numeric (no?)

Need to see how it performs with a very large amount of data

# Some other input from Brian

- Would like to see Keywords on more entities (e.g. Instrument, XxxParameter, maybe Dataset)
- Define controlled vocabulary and use it for keywords

This will be more effort for the users. I prefer to see it after the free text search has been tried.

Input from PanData on what they need in controlled vocab, synonyms, keywords etc. would be useful for after 4.3

- Define an RDF format for the metadata model
- Possible sub-group to look at XML renderings of the ICAT schema as well as RDF

Harmonizing the XML work would be very useful

# Where next?

- I plan to start work on 4.3 very soon
  - please say ASAP if you are not happy with the planned contents

- Main thing from my perspective for the version beyond 4.3 is to improve the query capability
  - particularly to allow more complex rules
  - will try to maintain backwards compatibility