ICAT API Tutorial

Richard Tyer and Glen Drinkwater

13th August 2008

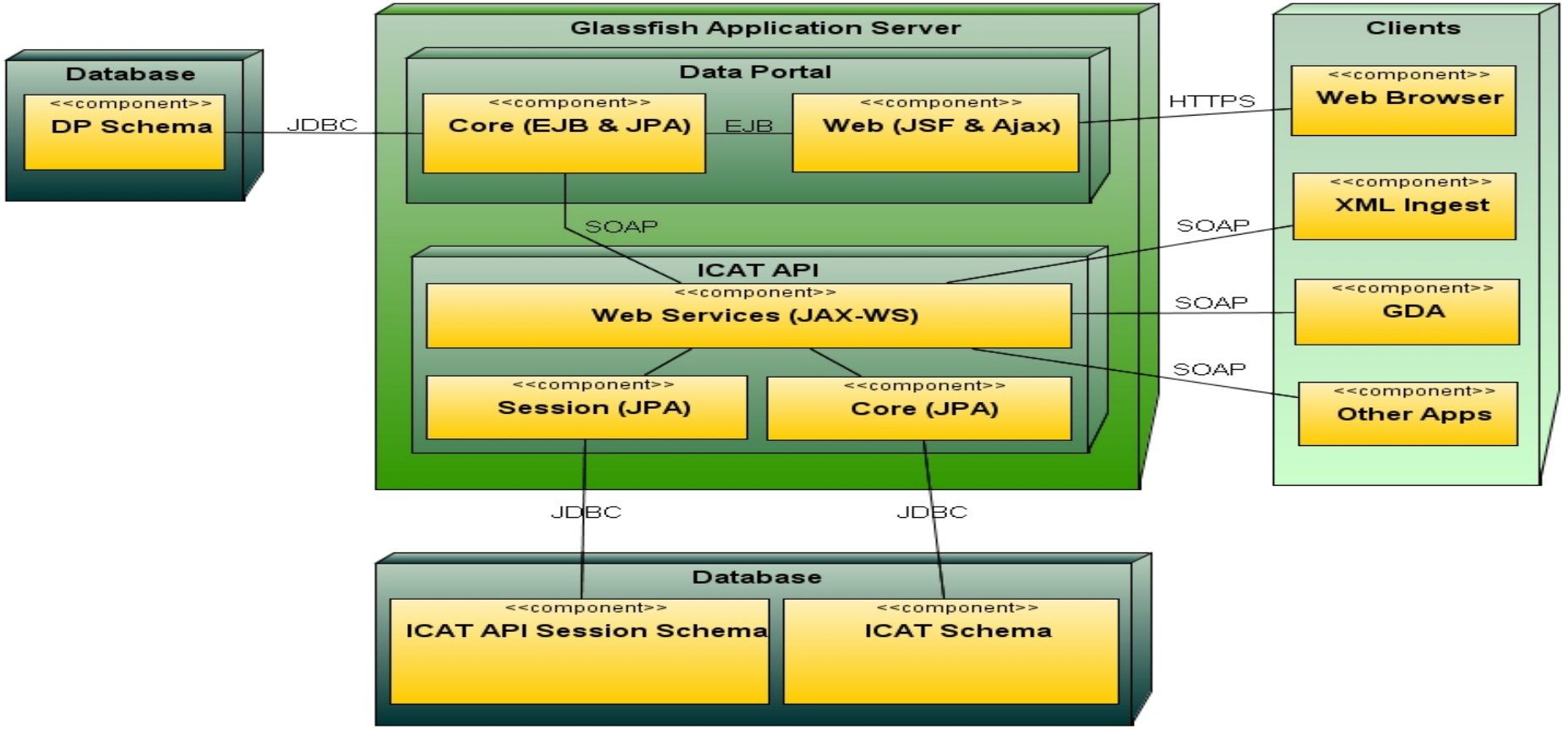
Introduction

- ICAT API provides uniform access for all applications wishing to use ICAT
  - Previously applications used their own ad-hoc solutions to interact with the ICAT.
  - Insulates clients from changes in the ICAT schema
- ICAT API offers a simple abstraction layer from the ICAT domain model
  - Compared to the complicated business rules and schema
  - Shields the developer from model, rules, and schema
- Allows interoperable with other languages and platforms
  - Can be used by any language supporting WS-I Basic Profile
- Extensibility - modular architecture combination of components for new services

ICAT API Overview

- Technologies and Standards
  - Java Enterprise Edition 5 (JEE5)
    - Java Persistence API – for interaction with databases
    - EJB3 – for business logic
    - JAX-WS – WS-I Basic Profile 1.1 for interoperability
      - .NET, Java, Perl and Python ICAT API clients
  - Glassfish Application server
    - Free, open source application server which implements JEE5
    - ICAT was developed against Glassfish but will work in any JEE5 application server

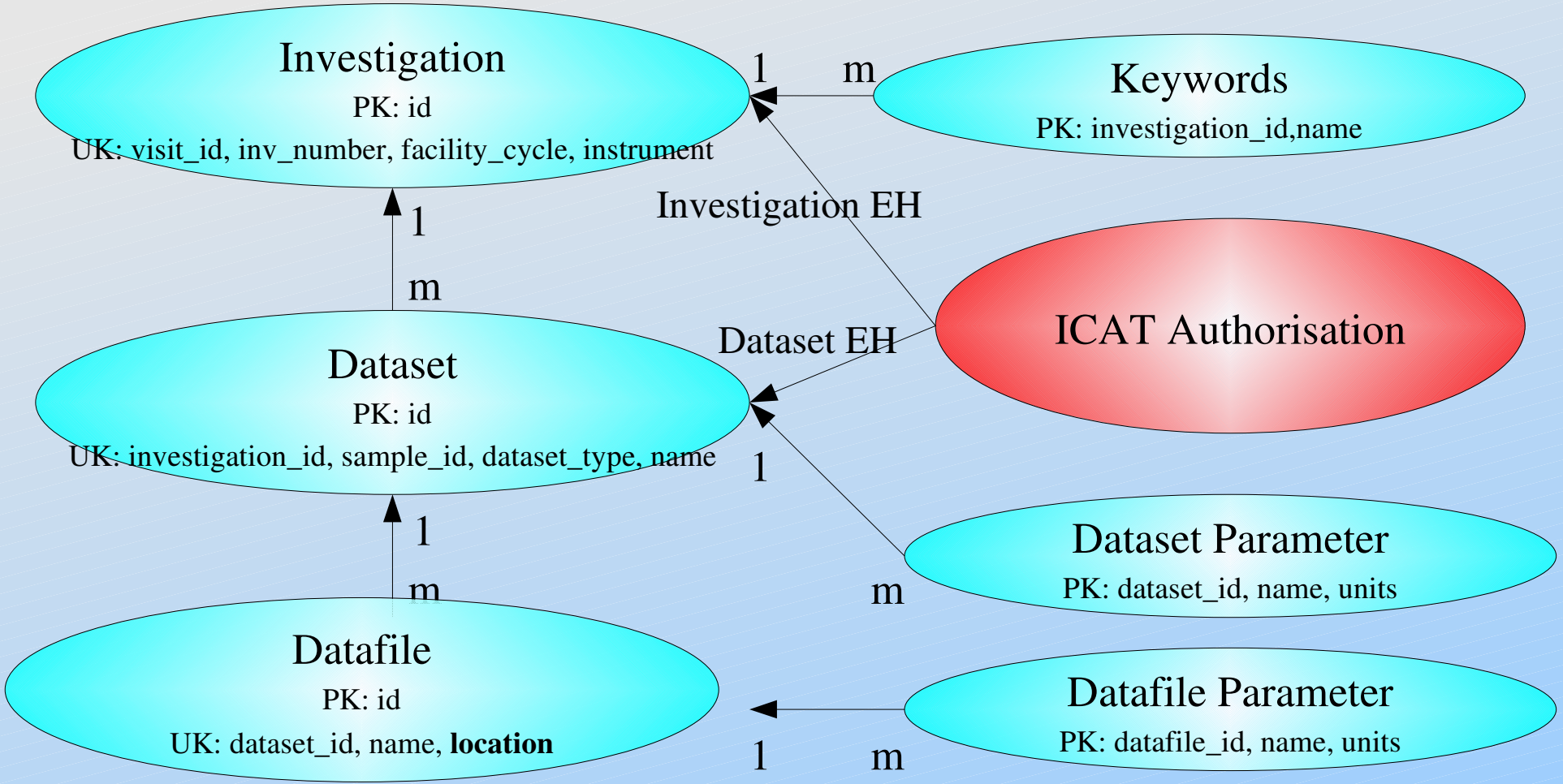
ICAT API Architecture



ICAT Overview

- Contains information about experiments and the data they produce.
- Based on the STFC Core Scientific Metadata Model v2 with many extensions
  - http://epubs.cclrc.ac.uk/work-details?w=30324
- Investigation->dataset->datafile orientated
- Indexing using keywords and parameters
- Constrains parameters building block of generic abilities, but also open to abuse - GIGO!
- Involved but efficient Authorisation framework taking various authorisation user stories into account
  - https://esc-cvs.dl.ac.uk/svn/dl/metadata/icat/trunk/documentation/authorisation/icat3_authorisation_spec.doc

Simplified ICAT Schema



Typical ICAT API workflow

- Compile WSDL to generate remote proxy and classes for business delegates
- Call remote proxy methods to interact with ICAT API
- Call business delegates to interrogate results returned from web service calls

Compiling WSDL

- Web Service Description Language is a XML schema used to describe a web service API
- WSDL can be compiled to generate:
  - classes for JAXB value types and exceptions
  - stub methods for WS service endpoint interfaces
- Can use command line tool, ant task or IDE for RAD

http://java.sun.com/javase/6/docs/technotes/tools/share/wsimport.html

ICAT Remote Proxy

- Remote proxy - Provides a reference to an object located in a different address space
- Key objects:
  - uk.icat3.client.ICATService – factory for service endpoint proxy
  - uk.icat3.client.ICAT – service endpoint proxy

```
ICATService service = new ICATService();
        ICAT icat = service.getICATPort();
```

ICAT User Login

- ICAT uses the SSO MyProxy, so authentication for users is via federal ID and password:

String login(String username, String password)
String loginLifetime(String username, String password, int lifetime)
boolean logout(String sessionId);

- Login web services return a session ID string, which is then used in subsequent web service calls

Investigation Keywords

- Investigations can be described by an arbitrary number of non-constrained keywords
- Can get a list of all keywords in use within own investigations or all keywords in the entire catalogue:

```
Collection<String> getKeywordsForUser(String sessionId)
Collection<String> getAllKeywords(String sessionId, KeywordType type)
```

Investigation Keyword Searches

- Can search for investigations containing set of keywords
- If multiple keywords specified, then matching investigations must be labelled with all keywords
- By default, keyword searches are case insensitive

```
Collection<Investigation> searchByKeywords(String sessionId,
                                           Collection<String> keywords)
```

KeywordDetails

- Class uk.icat3.client.KeywordDetails allows for more control over keyword searches within the ICAT
- Can control case sensitivity:

```
boolean isCaseSensitive()
void setCaseSensitive(boolean value)
```

- Use more powerful search method within ICAT API:

```
searchByKeywordsAll(String sessionId,  KeywordDetails kd,
                    int startIndex, int numberOfResults)
```

InvestigationInclude

- KeywordDetails has getters and setters for InvestigationInclude object
- InvestigationInclude is used to control how much of the data contained within an investigation is returned from the database and used to populate the business delegates
- Retrieving all information relating to an investigation is VERY expensive!

Searching using InvestigationInclude

- Valid values:

INVESTIGATORS_ONLY
KEYWORDS_ONLY
PUBLICATIONS_ONLY
INVESTIGATORS_AND_KEYWORDS
INVESTIGATORS_AND_SHIFTS
INVESTIGATORS_SHIFTS_AND_SAMPLES
INVESTIGATORS_SHIFTS_SAMPLES_AND_PUBLICATIONS
DATASETS_AND_DATASET_PARAMETERS_ONLY
DATASETS_ONLY
DATASETS_AND_DATAFILES
DATASETS_DATAFILES_AND_PARAMETERS
SAMPLES_ONLY
ROLE_ONLY
SHIFT_ONLY
ALL_EXCEPT_DATASETS_AND_DATAFILES
ALL
NONE
ALL_EXCEPT_DATASETS_DATAFILES_AND_ROLES

```
kd.setInvestigationInclude(InvestigationInclude.KEYWORDS_ONLY);
```

Retrieving a specific investigation

- Given the investigation ID one can retrieve a specific investigation:

```
Investigation getInvestigation(String sessionId, Long investigationId)

Investigation getInvestigation(String sessionId, Long investigationId,
                               InvestigationInclude includes)

Collection<Investigation> getInvestigations(String userId,
                                            Collection<Long> investigationIds,
                                            InvestigationInclude includes)
```

Retrieving specific datasets / data files

- If have the numerical ID for specific entities then do not need to retrieve the parent investigation(s)

Dataset getDataset(String SID, Long datasetID)
Datafile getDatafile(String SID, Long datafileID)

Drilling down into an investigation

- Once have retrieved an investigation, one can get a list of all the datasets that it contains:

```
List<Dataset> datasetList = someInvestigation.getDatasetCollection();
```

- Once have retrieved a dataset, one can get a list of all the data files that it contains:

```
List<Datafile> datafileList = someDataset.getDatafileCollection();
```

Downloading Data

- Datasets and data files can be downloaded via HTTP
- Use of HTTP facilitates client development and provides abstraction from underlying storage framework

```
String downloadDatafile(String SID, Long datafileID);
String downloadDatafiles(String SID, List<Long> datafileIDs);
String downloadDataset(String SID, Long datasetID);
```

ICAT Permissions Model

Role	Can remove?	Can modify?	Can download?	Can view?
Creator	Y	Y	Y	Y
Updater	N	Y	Y	Y
Downloader	N	N	Y	Y
Reader	N	N	N	Y

- Explicit permissions on investigation and datasets
- Implicit permissions on data files

Querying ICAT authorisation

- To get list of user role privileges:

```
Collection<IcatAuthorisation>
    getAuthorisations(String sid, Long elementId, ElementType elementType)
```

- Specify element type using ElementType enum:
  - ElementType.INVESTIGATION
  - ElementType.DATASET
  - ElementType.DATAFILE
- Returns collection of uk.icat3.client.IcatAuthorisation objects

Querying ICAT authorisation II

- Can extract user identities and roles from IcatAuthorisation objects:

```
String user = icatAuthorisation.getUserId();
IcatRole iRole = icatAuthorisation.getRole();
String role = iRole.getRole();
```

- uk.icat3.client.iRole has various getter methods to check permissions:
  - boolean isActionSelect()
  - boolean isActionInsert()
  - boolean isActionDownload()
  - boolean isActionUpdate()
  - boolean isActionDelete()
  - ...

Adding datasets and data files

- Need to have creator role on an investigation in order to create a dataset:

```
Dataset createDataSet(String sessionId, Long investigationId, Dataset dataSet)
Collection<Dataset> createDataSets( String sessionId,
                                    Collection<Dataset> dataSets,  Long investigationId)
```

- Need to have creator role on a dataset in order to create a datafile:

```
Datafile createDataFile(String sessionId,  Datafile dataFile, Long datasetId)
Collection<Datafile> createDataFiles( String sessionId,
                                      Collection<Datafile> dataFiles,  Long datasetId)
```

Setting authorisation within ICAT

- Can only grant permissions to level up to or including those that user holds on an entity

```
IcatAuthorisation
    addAuthorisation(String sid, String toAddUserId, String toAddRole,
                     Long elementId, ElementType elementType)
```

- Can get list of valid roles within ICAT:

```
Collection<IcatRole> listRoles(String sessionId)
```

- Includes:

CREATOR, DELETER, DOWNLOADER, READER, UPDATER

ICAT API Exceptions

- Limited number of exceptions thrown by ICAT API
  - uk.icat3.exceptions.InsufficientPrivilegesException
  - uk.icat3.exceptions.NoSuchObjectFoundException
  - uk.icat3.exceptions.SessionException
  - uk.icat3.exceptions.ValidationException
- ValidationException thrown if operation would violate a constraint within the ICAT database

ICAT Admin Web Service API

- Used by applications that need to act on behalf of other users
- Call loginAdmin() web service in admin WS API in order to get session ID for user. This session ID is then used with standard ICAT API.

  String loginAdmin(String runAsUserFedId)

- Applications are authenticated using HTTP Basic Authentication

Hands on Portion

- Now you will get a chance to try out the ICAT API for yourselves!
- Will step through a simple scenario:
  1. Set up prerequisite tools on laptops / machines
  2. Compile the ICAT API WSDL
  3. Authenticate with ICAT API to obtain session ID
  4. Perform some keyword searches to find investigations of interest
  5. Explore a couple of investigations - component datasets / data files
  6. Obtain URL for some datasets and data files and download these

Questions?