HZB Helmholtz Zentrum Berlin

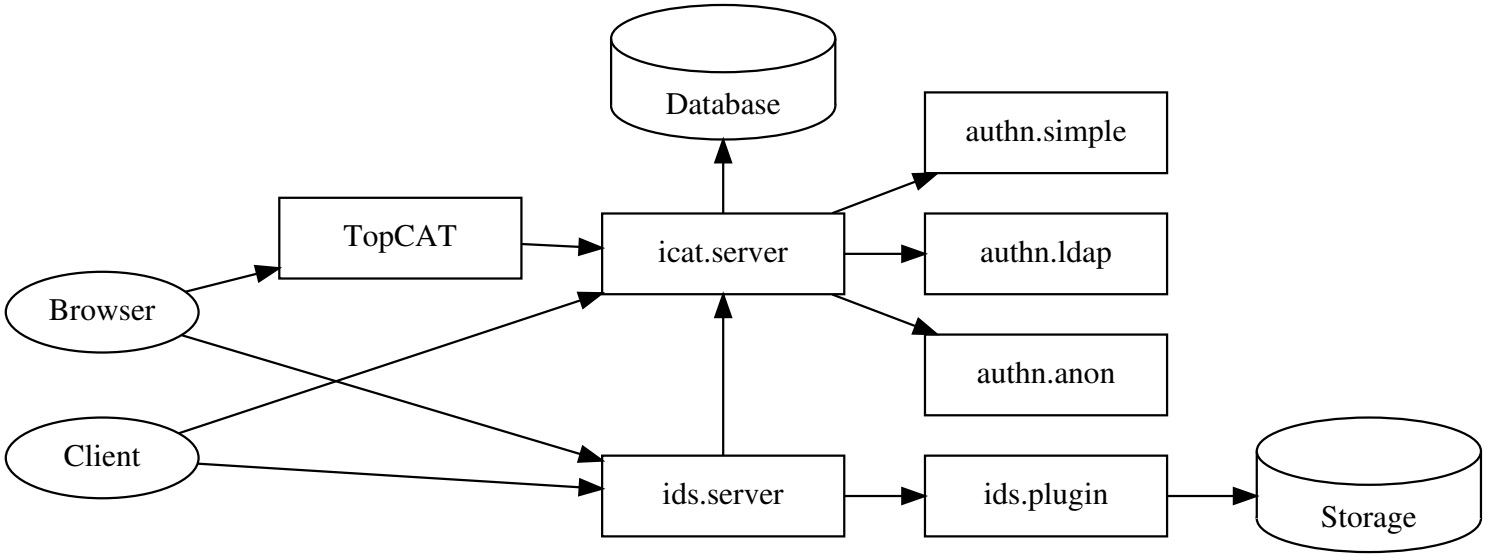
The ICAT component for OAI-PMH

Rolf Krahl

ExPaNDS workshop on EOSC, 6 Apr 2021

ICAT

- ICAT is a metadata catalogue. The main component is basically an API layer on a RDBMS.
- Java EE application, deployed in a Payara application server.
- Multiple components (just to name the most important):
  - `icat.server`: main component, the metadata catalogue.
  - `ids.server`: manages the storage, provides access (HTTP upload and download) to data files.
  - `topcat`: web user interface.



Challenges with OAI-PMH

- An OAI-PMH component needs to map metadata in the catalogue onto metadata standards requested by the harvester.
- Main challenge: we have diversity on both ends, input and output.
- At input:
  - although there is a common ICAT schema, each facility uses it a little differently and applies slightly different semantics to the entity classes.
  - different types of objects may need to get exposed: investigation vs. study vs. dataset vs. data publication.
- For the output:
  - we need to support at least Dublin Core and DataCite metadata.
  - we want to be open for future metadata standards also covering physical metadata of the measurement.
- All that flexibility should be configurable, without need to rebuild the program.

Concept

- Implement `icat.oaipmh` as a separate component. It listens at a dedicated endpoint and connects to `icat.server` as a client, using the standard API.
- Use standard ICAT access rules to control what may be disseminated.
- To make the mapping of metadata flexible and configurable, proceed in three steps:
  1. In a configuration file, select the entity classes that shall be exposed, along with relevant attributes and related objects.
  2. Out of that, `icat.oaipmh` generates an internal XML representation of the data using a one-to-one mapping from the ICAT schema.
  3. Use XSLT provided as a separate configuration file to generate the output from that internal XML representation.
- Example XSLT files for the most relevant cases (mapping generic ICAT investigation onto Dublin Core and DataCite) are provided as a starting point to write facility specific customized versions.

Usage: a practical example

run.properties

```
# The metadata formats to be supported
metadataPrefixes  = oai_dc oai_datacite

oai_dc.xslt       = /path/to/oai_dc.xslt
oai_datacite.xslt = /path/to/oai_datacite.xslt

# Identifiers for metadata configuration
data.configurations = inv stud
data.inv.metadataPrefixes = oai_dc oai_datacite
data.stud.metadataPrefixes = oai_dc

# Relevant data objects and properties
data.inv.mainObject = Investigation
data.inv.stringProperties = summary doi title
data.inv.subPropertyLists = investigationUsers
data.inv.investigationUsers.stringProperties = role
data.inv.investigationUsers.subPropertyLists = user
data.inv.investigationUsers.user.stringProperties = \
    fullName givenName familyName orcidId affiliation
```

Features and Conclusions

Features of `icat.oaipmh`:

- Support sets.
- Support selective harvesting by datestamps and by sets.
- Support flow control using `resumptionToken`.
- No support of maintaining information about deleted records.
- Configurable to disseminate any object class present in ICAT.
- Support multiple metadata formats by virtue of on-the-fly transformation using XSLT.