A FAIR Digital Object Ecosystem Testbed 27.05.2021

Andreas Pfeil, Thomas Jejkal (KIT – SCC-DEM)
The Helmholtz Association and its Mission

- Association of German large scale research labs
- In 2020, more than 43,000 researchers
- Annual budget amounts to € 5 billion
- Solving major challenges facing society, science and the economy focusing on six research fields
Helmholtz ‘Information & Data Science’ Incubator

- Initiated in 2016 to network and strengthen the Association’s expertise and enormous stores of data
- Set up as long-term, bottom-up process to push forward Helmholtz’ digitalization strategy
- Funding for at least 10 years
- Identified five topics with strategic importance
‘Helmholtz Metadata Collaboration’ (HMC) Platform

- Facilitate the **discovery, access, machine readability, and reuse** of research data of the Helmholtz Association by metadata.

- **Support researchers in** (automatically) **describing** their data by means of a suitable, standard-compliant description with metadata.

- Provide comprehensive and **sustainable services**, consulting, information and tools for efficient metadata handling as a distributed and shared facility.

- **Metadata Hubs** to support researchers in research fields by **suggesting standards and implementing services**.
Working Package 2: FAIR Data Commons

- Heterogenous research data silos in research fields
  → What can they have in common?
- FAIR DOs identified as possible top-level commonality
- Facilitates compliance with the FAIR principles for research data
- Foster uniform access to metadata by providing standardized interfaces
- Support scientist with easy-to-use tools, generally applicable processes, and best practices including training and consultancy
- Closely aligns efforts with (inter-)national consensus, e.g., coming from RDA, EOSC or NFDI
Why? The Testbed in the context of HMC

- HMC sees FAIR DOs as a method to harmonize data interoperability within Helmholtz
- Demonstrate feasibility of the FAIR DO concept
  - Using a set of services/APIs
  - Runnable on everyday computers (using docker)
Objectives of the testbed

- Facilitate FAIR use cases around PIDs and PID Information Records
- Demonstrate common FAIR use cases. Currently:
  - Create a PID (with enforced validation)
  - Update the PID record (with enforced validation)
  - Retrieve a PID (through a PID-service-independent interface)
  - Communicate the PID creation/update to other services (Messaging)
  - Search for PIDs (proof-of-concept)
- Stimulate discussions in HMC and beyond
A side note: Resolving a PID

**PID**
21.T11998/0000-001A-3905-F

resolve

A “landing page for machines”

follow URL

**Record**

**Resource**

**digitalObjectType**
21.T11148/e83481d4b6467110e7c9

"21.T11148/ManuscriptPage"

**digitalObjectLocation**
21.T11148/bb4578129050b83046264

"https://repository.example.com/data_experiment.xml"

**kernelInformationProfile**
21.T11148/076759916209e5d62bd5

"21.T11148/301c6f04763a16f0f72a"

**version**
21.T11148/c692273deb2772da307f

"1.0.0"

**dateModified**
21.T11148/397d831a3a9d18eb52c

"2021-05-14T11:45:15:374+00:00"

**digitalObjectPolicy**
21.T11148/8074aed799118ac263ad

"placeholder/pid/pointing/to/policy/object"

**dateCreated**
21.T11148/aaf5fb4c72222e2d950a

"2021-05-14T11:45:15:374+00:00"

**etag**
21.T11148/92e2003f11a56800b3e47

"{"sha256sum": \"V\'sha256 c50624fd5dd2b9652b72e2d20...}"
A side note: Resolving a PID

- Resolving this PID: 21.T11998/0000-001A-3905-F
- Using the handle resolver
  [https://hdl.handle.net/](https://hdl.handle.net/)
- Results in

<table>
<thead>
<tr>
<th>Handle.Net®</th>
<th>Handle Values for: 21.T11998/0000-001A-3905-F</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Index</strong></td>
<td><strong>Type</strong></td>
</tr>
<tr>
<td>1</td>
<td>URL</td>
</tr>
<tr>
<td>2</td>
<td>21.T11148/8eb858ee0b12e8e463a5</td>
</tr>
</tbody>
</table>
| 4           | 21.T11148/709a23220f2c3d64d1e1 | 2019-12-10 12:27:48Z | Sea-Bird SBE 37-IM MicroCAT C-T Seismic Thermistor 
| 5           | 21.T11148/4eaec4bc0f1df68ab2a7 | 2019-12-10 12:27:48Z | ["Owner": "National Centre", "ownerContact": "louise.darrough", "ownerIdentifierValue": "https://vocab.nerc.ac.uk/org/ORG000009/", "ownerIdentifierType": "Organization"] |
| 6           | 21.T11148/1f3e82ddf0697a497432 | 2019-12-10 12:27:48Z | ["Manufacturer": "manufacturerName Scientific", "modelName": "SBE 37-IM", "manufacturerIdentifierValue": "http://vocab.nerc.ac.uk/org/ORG000009/", "manufacturerIdentifierType": "Organization"] |
| 7           | 21.T11148/55f8ebc805e65b5b71dd | 2019-12-10 12:27:48Z | "identifierValue": "https://hdl.handle.net F", "identifierType": "MeasuringInstruments" |
| 8           | 21.T11148/76ad9d0324302fe47dd | 2019-12-10 12:27:48Z | A high accuracy conductivity and temperature sensor designed for deployment inductive modem for real-time data transfer and data storage. |

Submit
Create a PID based on already hosted data

- Assuming data which is e.g. URL-referenceable
- Put together record information:
  - URL (or another data reference)
  - Profile
  - Additional information according to the profile
- Send the information to the PIT service
  - It will validate the information
  - and register a PID on success
  - Requires authorization in real PID systems (not in the sandboxed one)
Manipulate PID record information

- Similar to PID generation
- The information in an existing PID can be updated
  - Requires authorization in real PID systems (not in the sandboxed one)
  - The modified record will be validated before it will be persisted

```
# PID Service
Default: Sandbox PID Service
Configurable: Handle Service

# PIT Service
Resolve, create, update PIDs. Following PID Information Types WG Recommendations.

# PIT Validation
update

# Record Information
R
```

Read a PIDs record information

- Also called “Resolving a PID”
- Different PID services can be configured
- but the API will stay the same.
- Currently Implemented:
  - Sandboxed
    - Default in the testbed
    - No real PIDs, only resolvable using this instance of PIT service
  - Handle Service
Inform other systems about changes

- The PIT service will send messages about changes it made
- Receiver is a message broker
  - forwards the message to services which registered for the event topic
  - topic example: “record.update”
Search for PIDS using record information (proof-of-concept)

- Indexing service using the message broker approach
- **Ingests record information** into a search index (elastic search)
- Retrieve PIDS by searching record information
Search for PID(s) using content information (planned)

- In planning
- Use the machine actionability of FAIR DOs
- Retrieve and **ingest extracted data**
  - to put even more information into the index.
Let us start the testbed, create a record and use the search functionality.
https://github.com/kit-data-manager/testbed4inf
Summary

- Easily create, update, resolve and search PID records
  - Automatic validation
  - Automatic indexing
- Used to investigate FAIR Digital Objects as a commonality between research fields (still ongoing)
- Contact
  - via Github Issues (see download)
  - via e-mail: andreas.pfeil@kit.edu

Download & Instructions

https://github.com/kit-data-manager/testbed4inf