ePIC Project Updates and Discussion

Digital Objects - from RDA Results towards Implementation

Ulrich Schwardmann (GWDG)

Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen (GWDG)

Am Fassberg, 37077 Göttingen
ulrich.schwardmann [at] gwdg.de

05 March 2019, Philadelphia

ePIC
Persistent Identifiers for eResearch

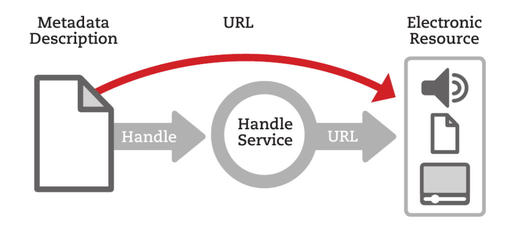
Content

1. Findability and PID
2. Reusability and Metadata
   - Linked Data
3. Interoperability and Registration of Types
   - Data Type Registries
   - Profiles and Policies
4. Accessibility & the DO Cloud
   - Techniques
   - Collections
   - DO Browser
   - Concepts
5. Searchability
6. Questions

Findability

Persistent Identification and Redirection

- URLs and cool URLs turned out to be highly instable
- PURL: persistent URLs, based on HTTP-redirection
  - central solution or administration/ownership unsolved
  - not reliable anymore by **organisational instability**
- better use **redirection** provided by a distributed system
- examples: URN, ARK, **Handle** (incl. DOI)



Handle-PIDs:

`21.11234/12345678`

`[Prefix]/[Suffix]`

Critical: Redirection always needs Resolution

The **ePIC** *Persistent Identifier Consortium for eResearch*

is a network of currently eight strong scientific service providers and a community infrastructure that signed a contract,

ePIC
Persistent Identifiers for eResearch
CSC
CSCS
Computer Network Information Center
Chinese Academy of Sciences
DKRZ
grnet
GWDG
SNIC
SURF SARA
CLARIN
Common Language Resources and Technology Infrastructure
ePIC
Persistent Identifiers for eResearch

- to **ensure a reliable and persistent identifier infrastructure**,
- devoted to the needs of the research community at large.
- Quality of Service
- **Mayor focus**: referability
  - for sharing data during the research process
  - with finer granularity and
  - PID coupled metadata

Reusability

needs Metadata

- needs knowledge about basic properties of data
  - **Metadata** is often unavailable, not connected to data or not interpretable
- For reuse provide as much of this knowledge before access to the data
  - Data Format Migration needs information about the format
- Registration:
  bind metadata and data with PID to a digital object

PID Information Types

- are additional metadata, stored in the PID database
- intended to be directly accessible without any redirection
- similar to mime types, typical examples are:
  - checksum
  - mime type
  - reference information
  - versioning (relative and absolute)
  - embargo time
  - expiration date
  - add. metadata location
  - basic Dublin Core
  - access restrictions and methods
  - data and table column formats
  - collection description
  - ...
- there will be more and others for IoT

Types vs. Linked Data

- An Example of a type: `isNextVersionOf`

This gives a triple:

- *pid-do1 type pid-do2*
- Digital-Object-1 `isNextVersionOf` Digital-Object-2

Thus one has a relation:

subject predicate object

with types as predicates.

- Types can be represented by PIDs again (DTR)

A feasability study at GWDG:

- mapping of type triples into a Neo4J graph database
- enables SPARQL queries
- realized as a Handle mirror with Neo4J database adapter

Types vs. Linked Data

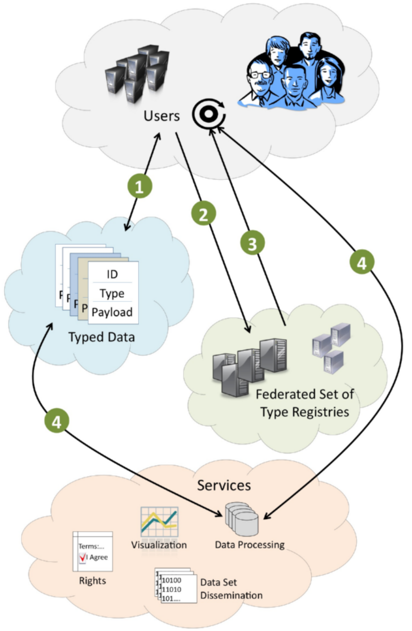
Examples of a types for metrology:

```
@prefix ePICdtr: <http://dtr.pidconsortium.eu/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://dtr.pidconsortium.eu/21.T11148/0a0fa93c89ac30e19d74>
ePICdtr:identifier <hdl:21.T11148/0a0fa93c89ac30e19d74> ;
ePICdtr:name "qty:time";
ePICdtr:properties "{’dimensions’:’T’,’name’:’unit:s’,
’issuer’:’BIPM’},’symbols’:{’alphabet’:’Latn’,’symbol’:’t’},
... }]}
ePICdtr:type ePICdtr:PID-BasicInfoType-Metrology .

<http://dtr.pidconsortium.eu/21.T11148/2f9571fa836af29bce01>
ePICdtr:identifier <hdl:21.T11148/2f9571fa836af29bce01> ;
ePICdtr:name "cnst:constant_Planck"; ...
```

Types vs. Linked Data

- currently only prototypical level
- required by customers to justify the choice of types
- Hierarchical Type Defintions lead to recursion in operation
  - which can be exploited automatically
- algorithm: Python with RDF plugin
- level of granularity still has to be determined

Interoperability by Registration of Types



RDA working group on
Data Type Registries

- approach to provide *type definitions*
- a PID for each definition
- defines the type structure, its use and semantics
- CORDRA as DTR service
- typical use cases:
  - with given PID find a type and ask for its use at DTR (see left)
  - ask at DTR for types with given semantics and find via PIDs according data

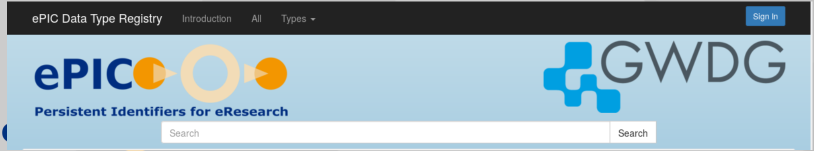
The ePIC Data Type Registry

- Features
  - Definition of PID Information Types
  - hierarchical types and automated schema extraction
  - Access via REST API, Browser
- based on CORDRA software
- GWDG is provider on behalf of ePIC
- Who can use the service?
  - public, authorization needed only for type definition

Overview: `http://dtr.pidconsortium.eu/`

PID InfoType states are:

- *in preparation* (21.T11148),
  - `http://dtr-test.pidconsortium.eu/`
- *candidate, approved, deprecated* (21.11104)
  - `http://dtr-pit.pidconsortium.eu/`



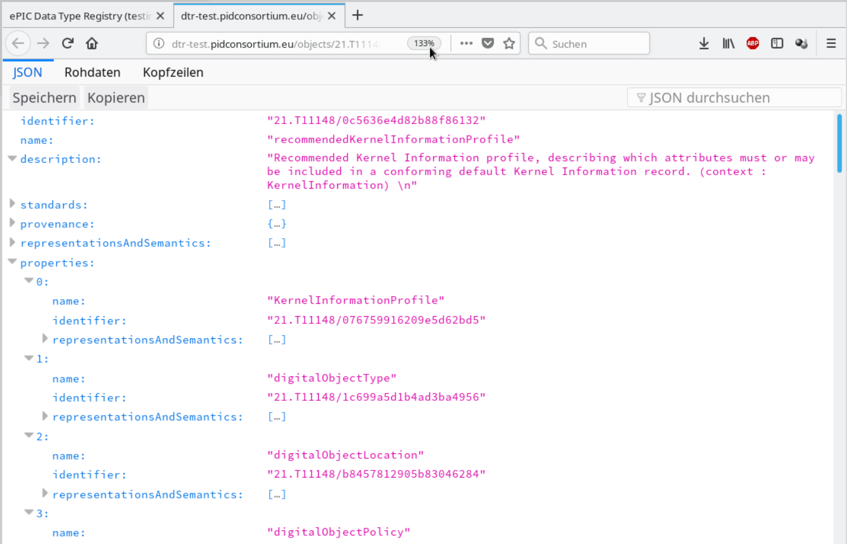
hierarchical type definitions

- types are often dependent from each other, how exactly?
- to exactly describe JSON objects by data types one needs:
  - a distinction between derived objects and basic objects
    - concept of *basic PID info types* and *PID info types*
  - a more exact description of the type dependencies
  - additionally a JSON schema inspired dependency model
- in consequence:
  - possibility to derive JSON schemas for the type values
    - automated server side schema derivation at ePIC DTR
  - one type defines in an exact way its whole dependencies
    - in objects of a certain type one can use the names of its parts (instead of type identifiers)
- see also Schwardmann, U.: Automated schema extraction for PID information types
  - PID: http://hdl.handle.net/21.11101/0000-0002-A987-7

Profiles and Policies
ePIC KernelInformationProfile

- created a *DTR Schema* **KernelInformationProfile** `21.T11148/532ce6796e2828dd2be6`
  - is a type registry schema (type) like PID-InfoType
- created an instance **recommendedKernelInformationProfile**: `21.T11148/076759916209e5d62bd5`
  - based on *DTR Schema* KernelInformationProfile
  - consisting of all recommended Kernel Information Types
- created an instance **KernelInformationPolicy**: `21.T11148/f9aa655f3c6cb14bd7b0`
  - objectLifeCycleType (M), objectTombstoneInformation (O), objectLicense (O)
  - objectLifeCycleType: currently Unicode-String, could be a controlled vocabulary: static, dynamic_irregular, dynamic_regular, ...

ePIC KernelInformationProfile


ePIC Data Type Registry (testi
dtr-test.pidconsortium.eu/obj
dtr-test.pidconsortium.eu/objects/21.T1114
133%
Suchen
JSON
Rohdaten
Kopfzeilen
Speichern
Kopieren
JSON durchsuchen

```
identifier:                       "21.T11148/0c5636e4d82b88f86132"
name:                             "recommendedKernelInformationProfile"
▼ description:                    "Recommended Kernel Information profile, describing which attributes must or may
                                  be included in a conforming default Kernel Information record. (context :
                                  KernelInformation) \n"
▶ standards:                      […]
▶ provenance:                     {…}
▶ representationsAndSemantics:    […]
▼ properties:
  ▼ 0:
      name:                       "KernelInformationProfile"
      identifier:                 "21.T11148/076759916209e5d62bd5"
    ▶ representationsAndSemantics: […]
  ▼ 1:
      name:                       "digitalObjectType"
      identifier:                 "21.T11148/1c699a5d1b4ad3ba4956"
    ▶ representationsAndSemantics: […]
  ▼ 2:
      name:                       "digitalObjectLocation"
      identifier:                 "21.T11148/b8457812905b83046284"
    ▶ representationsAndSemantics: […]
  ▼ 3:
      name:                       "digitalObjectPolicy"
```

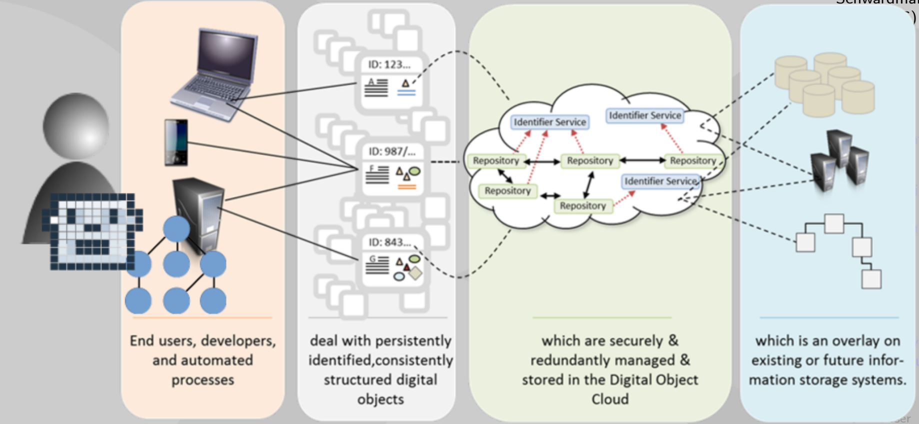
Persistent Identifiers for eResearch

How could a Policy look like

- Examples
  - suffix generator (counter, hash)
  - deletion allowed/forbidden
  - use of profile for information types
  - inheritage of profile elements from prefix to suffix
  - inheritage of policy elements from prefix to suffix
- all those can be described by boolean values or controlled vocabulary

Accessibility
and the Digital Object Cloud

The users view of the DO Cloud



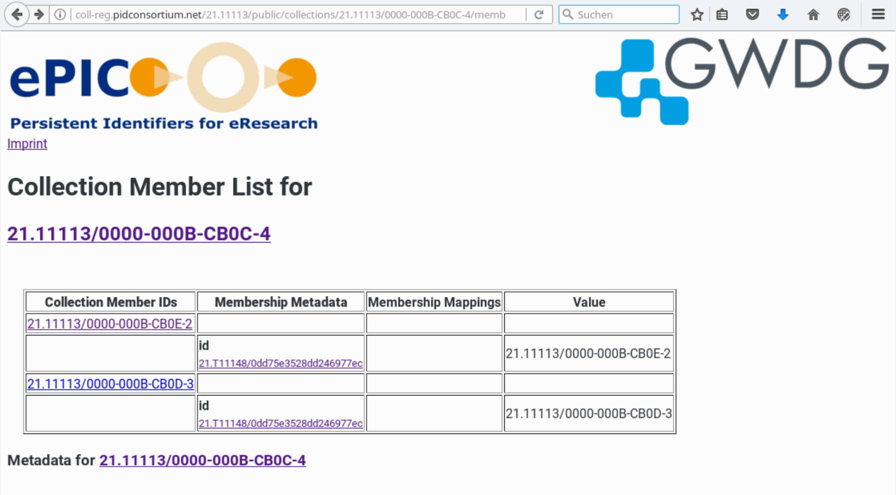
End users, developers, and automated processes deal with persistently identified,consistently structured digital objects which are securely & redundantly managed & stored in the Digital Object Cloud which is an overlay on existing or future information storage systems.[1]

[1] Global Digital Object Cloud, Larry Lannom, 2016

What are Collections in the RDA sense?

- Abstractly they are PIDs pointing to a list of PIDs
  - and additional metadata to enable services
  - this is a **recursive** definition: members can be collections
- the RDA outcome is a concrete REST API to manage collections
- collections are ubiquous also in data management:
  - directories, zip and tar archives, ...
  - objects structured by chapters, pages, newlines, ...
  - group definitions, ...
- collections are a very general way to organize objects hierarchically
  - PIDs are a completely flat view on global objects
  - the RDA collection helps to build hierarchies on objects
  - they only need **names as additional metadata** to make sense also for humans
- often repositories have an implicit hierarchical structure

A Collection Repository


coll-reg.pidconsortium.net/21.11113/public/collections/21.11113/0000-000B-CB0C-4/memb
Suchen
GWDG

ePIC

Persistent Identifiers for eResearch

Imprint

Collection Member List for

21.11113/0000-000B-CB0C-4

Collection Member IDs	Membership Metadata	Membership Mappings	Value
21.11113/0000-000B-CB0E-2			
	id 21.T11148/0dd75e3528dd246977ec		21.11113/0000-000B-CB0E-2
21.11113/0000-000B-CB0D-3			
	id 21.T11148/0dd75e3528dd246977ec		21.11113/0000-000B-CB0D-3

Metadata for 21.11113/0000-000B-CB0C-4

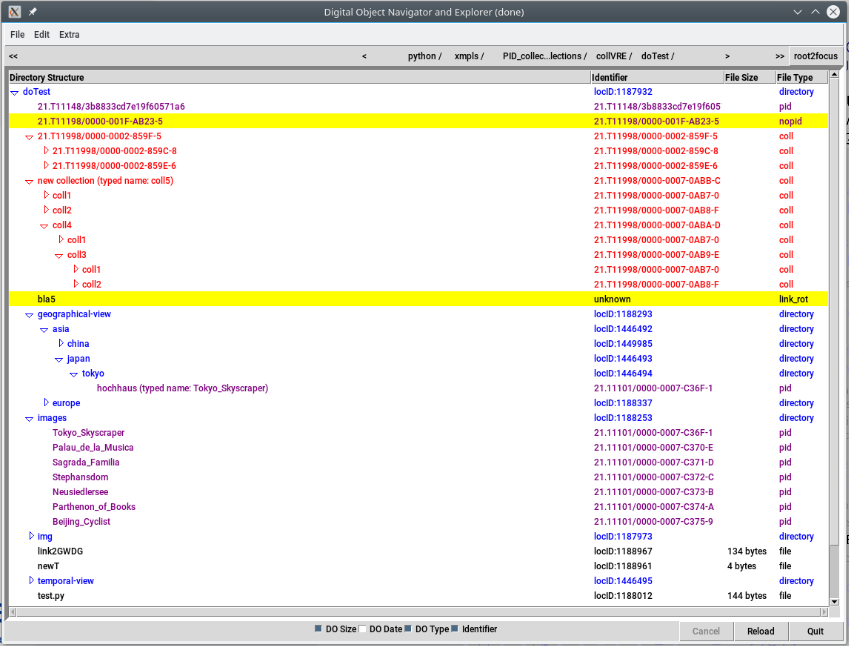
ePIC

Persistent Identifiers for eResearch

A Collection Repository

- a **repository agent** in the Digital Object Access Protocol
  - maintains a repository based on type entries in the collection PID
  - defines adaptor classes for different collection like structures
- or an **adaptor** into the Digital Object Access Protocol
  - DARIAH repository (humanities)
    - presents collections based on PIDs, but has no RDA collection API
  - IPCC-EFDB emission factor repository (climate research)
    - collection PIDs provided by ePIC Collection Repository
    - endpoints provided by ePIC PID service
  - ITIS taxonomy (biology)
    - based on unique and stable internal reference numbers
    - implementation via templates or fragment identifiers

A Digital Object Browser



Directory Structure	Identifier	File Size	File Type
▽ doTest	locID:1187932		directory
21.T11148/3b8833cd7e19f60571a6	21.T11148/3b8833cd7e19f605		pid
21.T11198/0000-001F-AB23-5	21.T11198/0000-001F-AB23-5		nopid
▽ 21.T11998/0000-0002-859F-5	21.T11998/0000-0002-859F-5		coll
▷ 21.T11998/0000-0002-859C-8	21.T11998/0000-0002-859C-8		coll
▷ 21.T11998/0000-0002-859E-6	21.T11998/0000-0002-859E-6		coll
▽ new collection (typed name: coll5)	21.T11998/0000-0007-0ABB-C		coll
▷ coll1	21.T11998/0000-0007-0AB7-0		coll
▷ coll2	21.T11998/0000-0007-0AB8-F		coll
▽ coll4	21.T11998/0000-0007-0ABA-D		coll
▷ coll1	21.T11998/0000-0007-0AB7-0		coll
▽ coll3	21.T11998/0000-0007-0AB9-E		coll
▷ coll1	21.T11998/0000-0007-0AB7-0		coll
▷ coll2	21.T11998/0000-0007-0AB8-F		coll
bla5	unknown		link_rot
▽ geographical-view	locID:1188293		directory
▽ asia	locID:1446492		directory
▷ china	locID:1449985		directory
▽ japan	locID:1446493		directory
▽ tokyo	locID:1446494		directory
hochhaus (typed name: Tokyo_Skyscraper)	21.11101/0000-0007-C36F-1		pid
▷ europe	locID:1188337		directory
▽ images	locID:1188253		directory
Tokyo_Skyscraper	21.11101/0000-0007-C36F-1		pid
Palau_de_la_Musica	21.11101/0000-0007-C370-E		pid
Sagrada_Familia	21.11101/0000-0007-C371-D		pid
Stephansdom	21.11101/0000-0007-C372-C		pid
Neusiedlersee	21.11101/0000-0007-C373-B		pid
Parthenon_of_Books	21.11101/0000-0007-C374-A		pid
Beijing_Cyclist	21.11101/0000-0007-C375-9		pid
▷ img	locID:1187973		directory
link2GWDG	locID:1188967	134 bytes	file
newT	locID:1188961	4 bytes	file
▷ temporal-view	locID:1446495		directory
test.py	locID:1188012	144 bytes	file

data driven research relies on **methods using data**

and **access to and management of data relies on operations on data**

- it is therefore even more important to have
  - **reliable references to operations**
  - and the **exakt description of operations**
- Technology for cross domain operations: **web services**
  - which are given by ressources (not operations) and methods (operations in operations)
- WSDL/RSDL tries to give descriptions for web services
- But the expressiveness of WSDL/RSDL is very limited
  - there is often no WSDL/RSDL at all necessary for REST
  - the operations are **only described by API descriptions**

data driven research relies on **methods using data**

and **access to and management of data relies on operations on data**

- it is therefore even more important to have
  - **reliable references to operations**
  - and the **exakt description of operations**
- Technology for cross domain operations: **web services**
  - which are given by ressources (not operations) and methods (operations in operations)
- WSDL/RSDL tries to give descriptions for web services
- But the expressiveness of WSDL/RSDL is very limited
  - there is often no WSDL/RSDL at all necessary for REST
  - the operations are **only described by API descriptions**

data driven research relies on **methods using data**

and **access to and management of data relies on operations on data**

- it is therefore even more important to have
  - **reliable references to operations**
  - and the **exakt description of operations**
- Technology for cross domain operations: **web services**
  - which are given by ressources (not operations) and methods (operations in operations)
- WSDL/RSDL tries to give descriptions for web services
- But the expressiveness of WSDL/RSDL is very limited
  - there is often no WSDL/RSDL at all necessary for REST
  - the operations are **only described by API descriptions**

data driven research relies on **methods using data**

and **access to and management of data relies on operations on data**

- it is therefore even more important to have
  - **reliable references to operations**
  - and the **exakt description of operations**
- Technology for cross domain operations: **web services**
  - which are given by ressources (not operations) and methods (operations in operations)
- WSDL/RSDL tries to give descriptions for web services
- But the expressiveness of WSDL/RSDL is very limited
  - there is often no WSDL/RSDL at all necessary for REST
  - the operations are **only described by API descriptions**

Try to make data operations simpler

Can we try to describe data operations similar to mathematical functions

$$f : S \to T, \; s \mapsto f(s) = t$$

where f is the function name, S (source) and T (target) are domain and codomain of both: data and metadata (incl. AAI)?

- Lets have a look at the definitions in the DOIP draft:
- *operation/function name*
  - `operationId`: is f , the identifier of the operation
- *data*
  - `targetId` (S): Id of the source DO
  - `input/output` (S,T): arbitrary I/O streams.
- *metadata*
  - `requestId` (S,T): the (unique) identifier of the request
  - `attributes` (S,T): optional array of JSON properties
  - `clientId` (S): the identifier of the client (AAI).
  - `authentication` (S): optional AAI JSON (sub) object
  - `status` (T): status identifier

Try to make data operations simpler

Can we try to describe data operations similar to mathematical functions

$$f : S \to T,\ s \mapsto f(s) = t$$

where f is the function name, S (source) and T (target) are domain and codomain of both: data and metadata (incl. AAI)?

- Lets have a look at the definitions in the DOIP draft:
- *operation/function name*
  - `operationId`: is f , the identifier of the operation
- *data*
  - `targetId` (S): Id of the source DO
  - `input/output` (S,T): arbitrary I/O streams.
- *metadata*
  - `requestId` (S,T): the (unique) identifier of the request
  - `attributes` (S,T): optional array of JSON properties
  - `clientId` (S): the identifier of the client (AAI).
  - `authentication` (S): optional AAI JSON (sub) object
  - `status` (T): status identifier

Searchability ???

Hasn't Google solved the searchability question?

- Searchability actually means *reverse lookup*
  - findability was answered by: get the data for the reference
  - searchability means: get the reference for some criteria
- this raises a lot of questions
  - technical implementation
    - centralized vs. distributed
    - scalability
    - access control
    - data base
    - query languages
  - legal, social
    - privacy
    - GDPR
  - governance and trust
  - ...

Many Thanks

Questions ???

Contact at ePIC:

- support [at] pidconsortium.eu

Contact at GWDG:

- **Ulrich Schwardmann**
  T: 0551 201-1542, E: ulrich.schwardmann [at] gwdg.de

ePIC Persistent Identifiers for eResearch